1972

Filter design using minimization techniques.

Maher Ahmed. Sid-Ahmed
University of Windsor

FILTER DESIGN USING MINIMIZATION TECHNIQUES

by

Maher Ahmed Sid Ahmed

A Thesis
Submitted to the Faculty of Graduate Studies through the
Department of Electrical Engineering in Partial Fulfillment
of the Requirements for the Degree of
Master of Applied Science at the
University of Windsor

Windsor , Ontario

1971

ABSTRACT

This work develops methods for choosing the coefficients of a filter transfer function, (digital or analog), to meet required magnitude specifications in the frequency domain.

The method developed make use of the unconstrained minimization technique of Fletcher and Powell[5], to minimize a magnitude square-error criterion in the frequency domain.

At the end of this work an extension is made to include phase specifications, as well as magnitude specifications.

ACKNOWLEDGMENTS

The writer wishes to express deep gratitude to his supervisor, Dr.G.Jullien , for his guidance and many helpful suggestions during the course of this study.

Thanks are due to Dr.M.Shridhar for his concern and valuable comments.

The financial support provided by the National Research Council of Canada is gratefully acknowledged.

TABLE OF CONTENTS

CHAPTER I

INTRODUCTION

1.1 LINEAR FILTERING

Linear filtering may be performed digitally by special or general-purpose computers as well as continuously by analog computers or RLC active or passive networks.

This work is aimed at designing digital and analog filters by implementing a minimization technique, to minimize a square error criterion between the magnitude of a transfer function and and a desired magnitude-frequency response. The technique used for minimization is a conjugate gradient method derived by Fletcher and Powell.

1.2 ON DIGITAL FILTERING

In this work we shall consider certain classes of frequency-selective linear discrete systems, or digital filters , as they are commonly called. Digital filters utilize digital components as the basic element for implementation. The trends toward decreased cost and size of digital components makes digital filtering especially attractive at this time. These trends promise to end the virtual monopoly of analog components for realizing real-time filters.

Since the first requirement for understanding the concept of the digital filter is a familiarity with difference equations, the next section introduces the subject.

1.3 THE DIFFERENCE EQUATION

The general mth-order difference equation is given by

$$y(nT)=\sum_{i=0}^{r}L_i\,x(nT-iT)-\sum_{i=1}^{m}K_i\,y(nT-iT) \qquad (1.1)$$

where, L_i and K_i are constants.

The form (1.1) shows the iterative nature of the difference equation; given the m previous output values y(nT-T), y(nT-2T), etc., and the r+1 most recent values of the input x(nT), x(nT-T), etc., the new output may be computed.

A mathematical tool for solving the difference equation is the z-transform.

1.4 DEFINITION OF THE z-TRANSFORM

Given a discrete-time signal f, represented by the sequence of numbers $\{f_n\}$, its z-transform F(z) is defined by the power series

$$F(z)=\sum_{n=0}^{\infty}f_n z^{-n} \qquad (1.2)$$

where z is a complex variable. The operation of taking the z-transform of a sequence will be denoted by :

$$F(z)=Z(f_n) \qquad (1.3)$$

The z-transform is a linear operation, so that

$$Z(af_n+bg_n)=aZ(f_n)+bZ(g_n) \qquad (1.4)$$

where a and b are constants.

An important property that can be derived from (1.2) is the z-transform of a delayed sequence that has zero value at negative time :

$$Z(f_{n-k})=z^{-k}Z(f_n) \qquad (1.5)$$

Equation (1.5) may be used to derive the important property of the z-transform :

$$F(z)=G(z)H(z) \tag{1.6}$$

This is represented in the time domain as the discrete convolution

$$f_n=\sum_{j=0}^{\infty}h_j g_{n-j} \tag{1.7}$$

where $F(z)=Z(f_n)$, $G(z)=Z(g_n)$, and $H(z)=Z(h_n)$.

1.5 SOLUTION OF THE mTH-ORDER DIFFERENCE EQUATION USING z-TRANSFORM

Take the z-transform of Eq.(1.1), and assume that the initial conditions are zero when the iteration begins.

Therefore,

$$Y(z)=\sum_{i=0}^{r}L_i X(z)z^{-i} -\sum_{i=1}^{m}K_i Y(z)z^{-i} \tag{1.8}$$

rewriting

$$Y(z)\sum_{i=0}^{m}K_i z^{-i}=X(z)\sum_{i=0}^{r}L_i z^{-i} \tag{1.9}$$

where $K_0=1$

or

$$Y(z)=X(z)\frac{\sum_{i=0}^{r}L_i z^{-i}}{\sum_{i=0}^{m}K_i z^{-i}} =X(z)H(z) \tag{1.10}$$

Thus the ratio of the z-transform of y_n, and the z-transform of x_n is a system function H(z) which is a rational fraction in z^{-1},and is a function of the constant coefficients in the original difference equation.

From (1.10) we can derive the canonic form of the mth-order difference equation. Defining an intermediate sequence e(nT) with associated z-transform E(z) such that

$$E(z)=\frac{X(z)}{\sum_{i=0}^{m}K_i z^{-i}} \tag{1.11}$$

$$Y(z) = E(z)\sum_{i=0}^{r} L_i z^{-i} \tag{1.12}$$

From (1.11) $$E(z) = X(z) - E(z)\sum_{i=1}^{m} K_i z^{-i} \tag{1.13}$$

Eq.(1.13) and (1.12) are used to realize the canonic form shown in fig.(1.1).

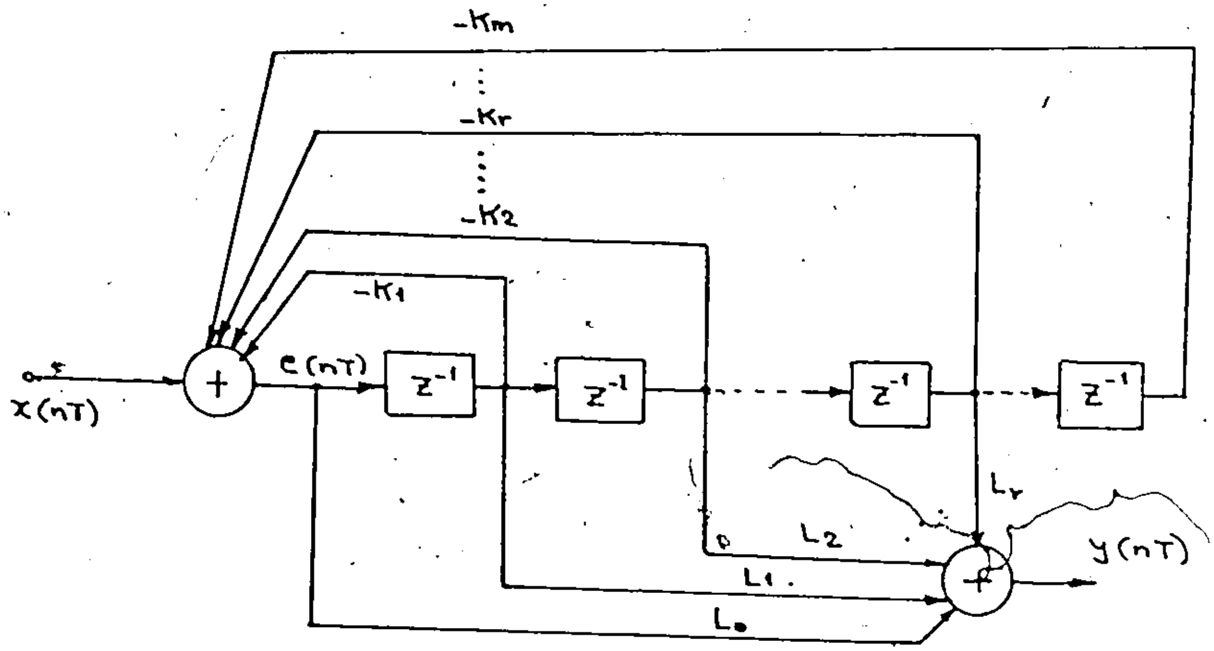


Fig.(1.1) canonic form of the mth-order network.

1.6 INTERPRETATION OF H(z) AS A FREQUENCY SELECTIVE FUNCTION

Assume that the input x(nT) is a sampled complex exponential wave

$$x(nT) = e^{jnwT} \tag{1.14}$$

The solution for y(nT) in Eq.(1.1) for such an input can be written

$$y(nT)=F(e^{jwT}).e^{jnwT} \tag{1.15}$$

Therefore, substituting in Eq.(1.1) we get

$$F(e^{jwT})=\frac{\sum_{i=0}^{r} L_i e^{-jiwT}}{\sum_{i=0}^{m} K_i e^{-jiwT}} = H(e^{jwT}) \tag{1.16}$$

Therefore , when dealing with the frequency domain

$$z=e^{jwT} \tag{1.17}$$

1.7 STABILITY OF A DIGITAL FILTER

Consider a Laurent expansion (ref.8) of a function F(z) about some point $z=z_o$

$$F(z)=\sum_{n=-\infty}^{\infty} a_n(z-z_o)^n \tag{1.18}$$

where

$$a_n=\frac{1}{2\pi j}\oint \frac{F(z)}{(z-z_o)^{n+1}}\,dz \tag{1.19}$$

From the definition of the z-transform

$$H(z) =\sum_{n=0}^{\infty} h(nT).z^{-n} \tag{1.20}$$

If we consider this to be a Laurent expansion of the function H(z) about the origin then

$$h(nT)=\frac{1}{2\pi j}\oint \frac{H(z)}{z^{-n+1}}\,dz \tag{1.21}$$

where c is the contour that encloses all the poles of $H(z)z^{n-1}$ and encircles the origin.

Eq.(1.21) is used at this point to show the significance of pole location on the time response . Consider a filter with a simple pole at z=a. Thus we can write

$$H(z) = \frac{H_1(z)}{z - a} \quad (1.22)$$

where $H_1(z)$ is assumed to contain no zeros or poles at $z=a$. By the theorem of residues

$$\frac{1}{2\pi j} \oint \frac{H_1(z)}{z - a} z^{n-1} \, dz = \sum \text{residues of } \frac{H_1(z)}{z - a} z^{n-1} \quad (1.23)$$

Now, the residue at $z=a$ is given by

$$H_1(a) a^{n-1} \quad (1.24)$$

From Eq.(1.24) the various modes of response as a function of the pole location can be examined . Since, in general, 'a' may be complex , let the pole be described by $a = |a| e^{j\phi}$ .Eq.(1.24) then becomes

$$H_1(a) |a|^{n-1} e^{j(n-1)\phi} \quad (1.25)$$

Note that the response time of Eq.(1.24) is an unbounded increasing function of n for $|a| > 1$, and a decaying function for $|a| < 1$.

From this we infer that H(z) is stable if all of its poles lie inside the unit circle.

CHAPTER II

AN OUTLINE OF FLETCHER AND POWELL DESCENT METHOD FOR MINIMIZATION

2.1 INTRODUCTION

A general function f(X) behaves as a pure quadratic function in the vicinity of its minimum . This statement is proved by considering a Taylor series expansion about the minimum X^*. Since the first partial derivatives of f(X) vanish at the minimum X^*, therefore

$$f(X)= f(X^*) + \tfrac{1}{2}(X-X^*)' A_f(X^*)(X-X^*) \qquad (2.1)$$

where $A_f(X)$, the matrix of second partial derivatives of f(X) evaluated at X^*, is positive definite.

Hence, it follows that the only methods which minimize a general function quickly and efficiently are those which (1) work well on a quadritic function and (2) are guaranteed to converge eventually for a general function.

2.2 CONJUGATE DIRECTIONS

A set of n independent and non-zero directions, $s_0, s_1, \ldots\ldots, s_{n-1}$ are said to be conjugate if , given a symmetric positive definite matrix A , they satisfy

$$s_i' A s_j = 0 \qquad 0 \leq i \neq j \leq n \qquad (2.2)$$

and, of course $$s_i' A s_i \neq 0 .$$

When using conjugate directions in minimizing routines a quadratic function of n variables can be minimized in n-steps.

The general quadratic function can be written

$$q(X) = a + b' AX + \tfrac{1}{2}X' AX \tag{2.3}$$

where A is positive definite and symmetric. Let X^* minimize $q(X)$. Then

$$\nabla q(X^*) = b + AX^* = 0 \tag{2.4}$$

Given a point X_o and a set of linearly independent directions $(s_o, s_1, \ldots\ldots, s_{n-1})$, constants B_i can be found such that

$$X^* = X_o + \sum_{i=0}^{n-1} B_i s_i \tag{2.5}$$

If the directions s_i are A-conjugate, and none are zero, then B_i can be determined from (2.5) as follows :

$$s_j' AX^* = s_j' AX_o + \sum_{i=0}^{n-1} B_i s_j' A s_i \tag{2.6}$$

Using (2.2), $\quad s_j' AX^* = s_j' AX_o + B_j s_j' A s_j$

and, using (2.4)

$$B_j = -(b + AX_o)' \frac{s_j}{s_j' A s_j} \tag{2.7}$$

Now consider an iterative minimization procedure, starting at X_o and successively minimizing $q(X)$ down the directions $s_o, s_1, \ldots\ldots, s_{n-1}$, where these directions satisfy (2.2). Successive points are then determined by the relation

$$X_{i+1} = X_i + \alpha_i s_i \qquad i=0,1,\ldots\ldots,n-1 \qquad (2.8)$$

where 'i' is the i^{th} iteration, and α_i is determined by minimizing $f(X_i + \alpha_i s_i)$, so that

$$s_i' \nabla q(X_{i+1}) = 0 \qquad (2.9)$$

Using (2.4) in (2.9) gives

$$s_i' (b + A(X_i + \alpha_i s_i)) = 0 \qquad (2.10)$$

or,

$$\alpha_i = - (b + AX_i)' \frac{s_i}{s_i' A s_i} \qquad (2.11)$$

From (2.8),

$$X_i = X_o + \sum_{j=0}^{i-1} \alpha_j s_j \qquad (2.12)$$

so that

$$X_i' A s_i = X_o' A s_i + \sum_{j=0}^{i-1} \alpha_j s_j' A s_i = X_o' A s_i \qquad (2.13)$$

Thus (2.11) becomes

$$\alpha_i = - (b + AX_o)' \frac{s_i}{s_i' A s_i} \qquad (2.14)$$

which is identical to (2.7) .Hence this sequential process leads, in n steps, to the minimum X^*.

2.3 METHOD OF FLETCHER AND POWELL

This method is designed so that a quadratic function of n variables is minimized in n iterations.

The method is summarized below:

Assume the function to be minimized is f(X), where X is the vector of the variables.

1- Choose an initial point X_o.

2- Form an identity matrix H_o.

3- Calculate the direction of search s_i from :

$$s_i = -H \nabla f(X_i)$$

4- Choose $\alpha = \alpha_i$ by minimizing $f(X_i + \alpha s_i)$ with respect to α .

5- Calculate the step-length $\sigma_i = \alpha_i \cdot s_i$.

6- Calculate $X_{i+1} = X_i + \sigma_i$

7- Update the matrix H_i

$$H_{i+1} = H_i + A_i + B_i$$

where
$$A_i = \frac{\sigma_i \sigma_i'}{\sigma_i' y_i} \quad , \quad y_i = \nabla f(X_{i+1}) - \nabla f(X_i)$$

$$B_i = \frac{H_i y_i y_i' H_i}{y_i' H_i y_i}$$

8- Set i=i+1 and return to 3.

CHAPTER III

USE OF THE UNCONSTRAINED MINIMIZATION TECHNIQUE IN THE DESIGN OF RECURSIVE DIGITAL FILTERS.

3.1 INTRODUCTION

The purpose of this chapter is to describe a practical method , (which improves on the method proposed by Steiglitz, see sec.3.10), for choosing the coefficients of a recursive digital filter to meet arbitrary specifications of magnitude-frequency response.

The proposed method uses the optimization algorithm described by Fletcher and Powell to minimize a square-error criterion in the frequency domain.

In chapter I , it was proved that the general form of a recursive filter has the form

$$Y(z) = \frac{\sum_{i=0}^{r} L_i z^{-i}}{1 + \sum_{i=0}^{m} K_i z^{-i}} \qquad (3.1)$$

However, with this form no control can be exercised over the pole location in the z-plane. To remedy this disadvantage a cascade form is choosen.

$$Y(z) = A \prod_{k=1}^{K} \frac{(z-r_k e^{j\phi_k})(z-r_k e^{-j\phi_k})}{(z-p_k e^{j\theta_k})(z-p_k e^{-j\theta_k})} \prod_{j=1}^{L} \frac{z-a_j}{z-b_j} \qquad (3.2)$$

which could be written in the form

$$Y(z)= A \prod_{k=1}^{K} \frac{z^2-2r_k\cos\phi_k z+r_k^2}{z^2-2p_k\cos\theta_k z+p_k^2} \prod_{j=1}^{L} \frac{z-a_j}{z-b_j} \tag{3.3}$$

3.3 CHOICE OF THE CRITERION FUNCTION

Let the desired magnitude be described at a discrete set of frequencies $w_1,w_2,\ldots\ldots,w_M$, and let Y_i^d be the desired magnitude at frequency w_i.

Choose a square-error criterion

$$Q(\psi_i)=\sum_{i=1}^{M}\left(\left|Y(z_i,\psi_i)\right| - Y_i^d\right)^2 \tag{3.4}$$

where $\psi_i=(r_k,\phi_k,p_k,\theta_k;a_j,b_j;A)^T$ is the 4K+2L+1 vector of unknown coefficients, and

$$z_i = e^{jw_i\pi} \tag{3.5}$$

where w_i is given in fractions of the Nyquist rate.

3.4 ELIMINATION OF A AS AN UNKNOWN PARAMETER

Steiglitz ref.(1),has shown that it is possible to eliminate A , and hence decrease the number of unknown parameters.

To eliminate A from Q , define the 4K+2L dimensional vector

$$\psi = (r_k, \phi_k, p_k, \theta_k; a_j, b_j)^T \tag{3.6}$$

and write

$$Y(z,A,\psi) = A \prod_{k=1}^{K} \frac{z^2-(2r_k\cos\phi_k)z+r_k^2}{z^2-(2p_k\cos\theta_k)z+p_k^2} \prod_{j=1}^{L} \frac{z-a_j}{z-b_j} = AH(z,\psi) \tag{3.7}$$

Then

$$Q(A,\psi) = \sum_{i=1}^{M} \left(\left| AH(z_i,\psi) \right| - Y_i^d \right)^2 \tag{3.8}$$

Differentiate (3.8) with respect to A and set the result to zero to obtain the optimum value of A. Hence A^* (optimum A) is given by

$$\left| A^* \right| = \frac{\sum_{i=1}^{M} \left| H(z_i,\psi) \right| Y_i^d}{\sum_{i=1}^{M} \left| H(z_i,\psi) \right|^2} \tag{3.9}$$

Since the sign of A^* does not affect the magnitude characteristic , it will be taken as positive.

The Fletcher-Powell method is used to minimize the new error-criterion

$$Q(\psi) = Q(A^*,\psi) \tag{3.10}$$

3.5 CALCULATION OF THE GRADIENT OF Q WITH RESPECT TO ψ

The Fletcher-Powell method requires the calculation of the gradient of Q with respect to the parameter vector ψ .

Since, $$Q = \sum_{i=1}^{M} (A^* |H(z_i, \psi)| - Y_i^d)^2$$

Therefore,

$$\frac{\partial Q}{\partial \psi} = 2A^* \sum_{i=1}^{M} (A^* |H(z_i, \psi)| - Y_i^d) \frac{\partial |H(z_i, \psi)|}{\partial \psi} \qquad (3.11)$$

Writing

$$|H(z_i, \psi)| = (H(z_i, \psi)\overline{H(z_i, \psi)})^{\frac{1}{2}}$$

We have

$$\frac{\partial |H(z_i, \psi)|}{\partial \psi} = \frac{1}{H(z_i, \psi)} . \mathrm{Re}(\overline{H(z_i, \psi)} \frac{\partial H(z_i, \psi)}{\partial \psi}) \qquad (3.12)$$

3.6 STABILITY CONSIDERATIONS

For stability the poles should lie inside the unit circle of the z-plane , and to insure minimum phase constraint the zeros shall be constrained to lie also in the unit circle of the z-plane.

Therefore,

$$
\begin{aligned}
0 &\leq r_k \leq 1 \\
0 &\leq p_k \leq 1 \\
-1 &\leq a_j \leq 1 \\
-1 &\leq b_j \leq 1
\end{aligned}
\tag{3.13}
$$

To secure the above constraints , we choose a new set of variables , ref.(7), related to the above as follows

$$
\begin{aligned}
r_k &= \sin^2\alpha_k \\
p_k &= \sin^2\beta_k \\
a_j &= \sin\theta_{zj} \\
b_j &= \sin\theta_{pj}
\end{aligned}
\tag{3.14}
$$

Therefore,

$$
H(z,\psi) = \prod_{k=1}^{K} \frac{z^2-(2\sin^2\alpha_k\cdot\cos\phi_k)z+\sin^4\alpha_k}{z^2-(2\sin^2\beta_k\cdot\cos\theta_k)z+\sin^4\beta_k} \prod_{j=1}^{L} \frac{z-\sin\theta_{zj}}{z-\sin\theta_{pj}} \tag{3.15}
$$

3.7 CALCULATION OF $\dfrac{\partial|H(z_i,\psi)|}{\partial\psi}$

Since;
$$\frac{\partial|H(z_i,\psi)|}{\partial\psi}=\frac{1}{|H(z_i,\psi)|}\,\mathrm{Re}\left(\overline{H(z_i,\psi)}\,\frac{\partial H(z_i,\psi)}{\partial\psi}\right).$$

Therefore;

$$\frac{\partial|H(z_i,\psi)|}{\partial\alpha_k}=|H(z_i,\psi)|.\mathrm{Re}\left(\frac{-4\sin\alpha_k.\cos\alpha_k.\cos\phi_k.z_i+4\sin^3\alpha_k.\cos\alpha_k}{z_i^2-(2\sin^2\alpha_k.\cos\phi_k)z_i+\sin^4\alpha_k}\right)$$

$$\frac{\partial|H(z_i,\psi)|}{\partial\phi_k}=|H(z_i,\psi)|.\mathrm{Re}\left(\frac{2\sin^2\alpha_k.\sin\phi_k.z_i}{z_i^2-(2\sin^2\alpha_k.\cos\phi_k)z_i+\sin^4\alpha_k}\right).$$

$$\frac{\partial|H(z_i,\psi)|}{\partial\beta_k}=-|H(z_i,\psi)|.\mathrm{Re}\left(\frac{-4\sin\beta_k.\cos\beta_k.\cos\theta_k.z_i+4\sin^3\beta_k.\cos\beta_k}{z_i^2-(2\sin^2\beta_k.\cos\theta_k)z_i+\sin^4\beta_k}\right)$$

$$\frac{\partial H(z_i,\psi)}{\partial \theta_k} = -|H(z_i,\psi)|.\mathrm{Re}\left(\frac{2\sin^2{}_k.\sin{}_k.z_i}{z_i^2-(2\sin^2\beta_k.\cos\theta_k)z_i+\sin^4\beta_k} \right)$$

$$\frac{\partial |H(z_i,\psi)|}{\partial \theta_{zj}} = -|H(z_i,\psi)|.\mathrm{Re}\left(\frac{\cos\theta_{zj}}{z_i-\sin\theta_{zj}} \right)$$

$$\frac{\partial |H(z_i,\psi)|}{\partial \theta_{pj}} = -|H(z_i,\psi)|.\mathrm{Re}\left(\frac{\cos\theta_{pj}}{z_i-\sin\theta_{pj}} \right)$$

In order to use Fletcher-Powell subroutine (FMFP or DFMFP) supplied by IBM in the scientific subroutine package, another subroutine FUNCT(N,X,Q,DQ) is written in which N is the number of variables, X is the vector of the variables, Q is the criterion function and DQ is the gradient of the criterion function with respect to the parameter vector X. This subroutine is summarized in the next section.

3.8 SUMMARY OF THE SUBROUTINE FUNCT(N,X,Q,DQ)

1)Calculate H_i given by eq.(3.15) $i=1,2,\ldots\ldots,M$

2)Calculate A^* given by eq.(3.9)

3)Calculate

$$E_i = A^* |H_i| - Y_i^d \qquad i=1,2,\ldots\ldots,M$$

4)Calculate

$$Q = \sum_{i=1}^{M} E_i^2$$

5)Calculate

$$\frac{\partial |H_i|}{\partial \psi_n}, \quad n=1,2,\ldots,4K+2L$$

(Formulas are given in previous section)

6)Calculate

$$\frac{\partial Q}{\partial \psi_n} = 2A^* \sum_{i=1}^{M} E_i \frac{\partial |H_i|}{\partial \psi_n}, \qquad n=1,2,\ldots,4K+2L$$

This subroutine was written in double precision complex arithmetic in Fortran IV

3.9 EXAMPLES

Example 1 : A low-pass filter

Specification of the filter are

$w=0.0,0.09(0.01)$; $Y^d=1.0$

i.e. w is taken in steps of 0.01 from 0.0 to 0.09

$w=0.10$; $Y^d=0.5$

$w=0.11,0.20(0.01)$; $Y^d=0.0$

$w=0.21,1.0(0.1)$; $Y^d=0.0$

If optimization was started for K=1,L=1 at

$$\Psi_0=(0.4,0.4,0.4,0.4;0.4,0.4)^T$$

After 10 iterations

$Q=0.3659$

$A^*=0.6926$

Poles $\begin{pmatrix} 0.9092 \pm j0.2330 \\ 0.8735 \end{pmatrix}$

Zeros $\begin{pmatrix} 0.06557 \pm j0.03398 \\ -0.3188 \end{pmatrix}$

If we assume the filter to have the following form

$$Y(z)=A^*\frac{z^{-2}+a_1z^{-1}+b_1}{z^{-2}+c_1z^{-1}+d_1}\cdot\frac{1-a_{z1}z^{-1}}{1-b_{p1}z^{-1}}$$

Then,

$a_1=-0.13114$

$b_1=0.54550$

$c_1=-0.18183$

$d_1=0.8809$

$a_{z1}=-0.31877$

$b_{p1}=0.87348$

If now optimization is started for K=2 ,L=1, and Ψ_0 the same as before , then after 30 iterations

$Q=0.09887$

$A^*=0.5449$

$$\text{Poles}\ \begin{pmatrix} 0.90097 \pm j0.24487 \\ 0.89287 \pm j0.28659 \\ 0.87316 \end{pmatrix}$$

$$\text{Zeros}\ \begin{pmatrix} 0.93166 \pm j0.35903 \\ 0.601387 \pm j0.75623 \\ -0.6666 \end{pmatrix}$$

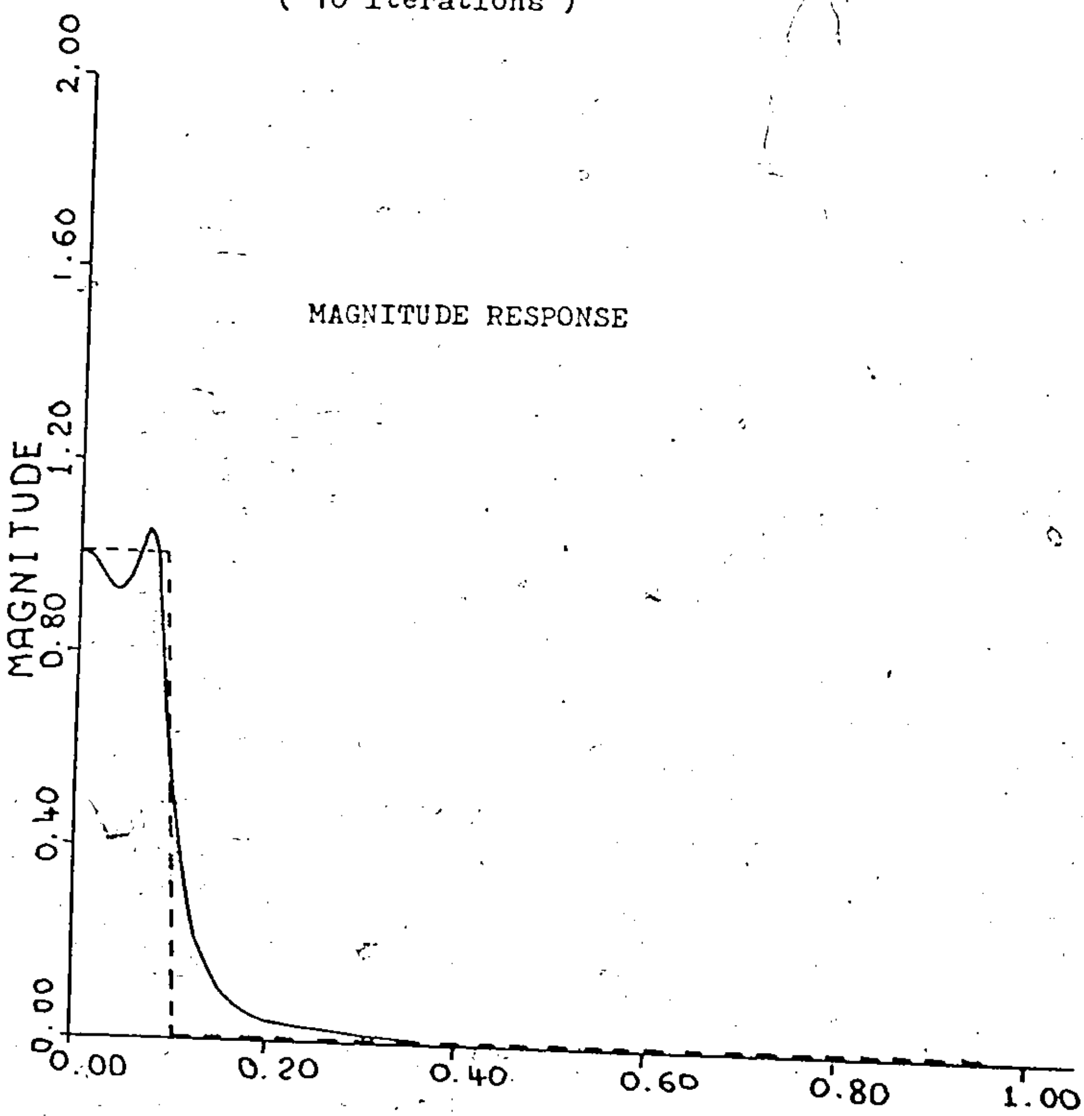
and if the transfer function is in the form

$$Y(z_1) = A^* \frac{1+a_1 z^{-1}+b_1 z^{-2}}{1+c_1 z^{-1}+d\ z^{-2}} \cdot \frac{1+a_2 z^{-1}+b_2 z^{-2}}{1+c_2 z^{-1}+d_2 z^{-2}} \cdot \frac{1-a_{z1} z^{-1}}{1-b_{p1} z^{-1}}$$

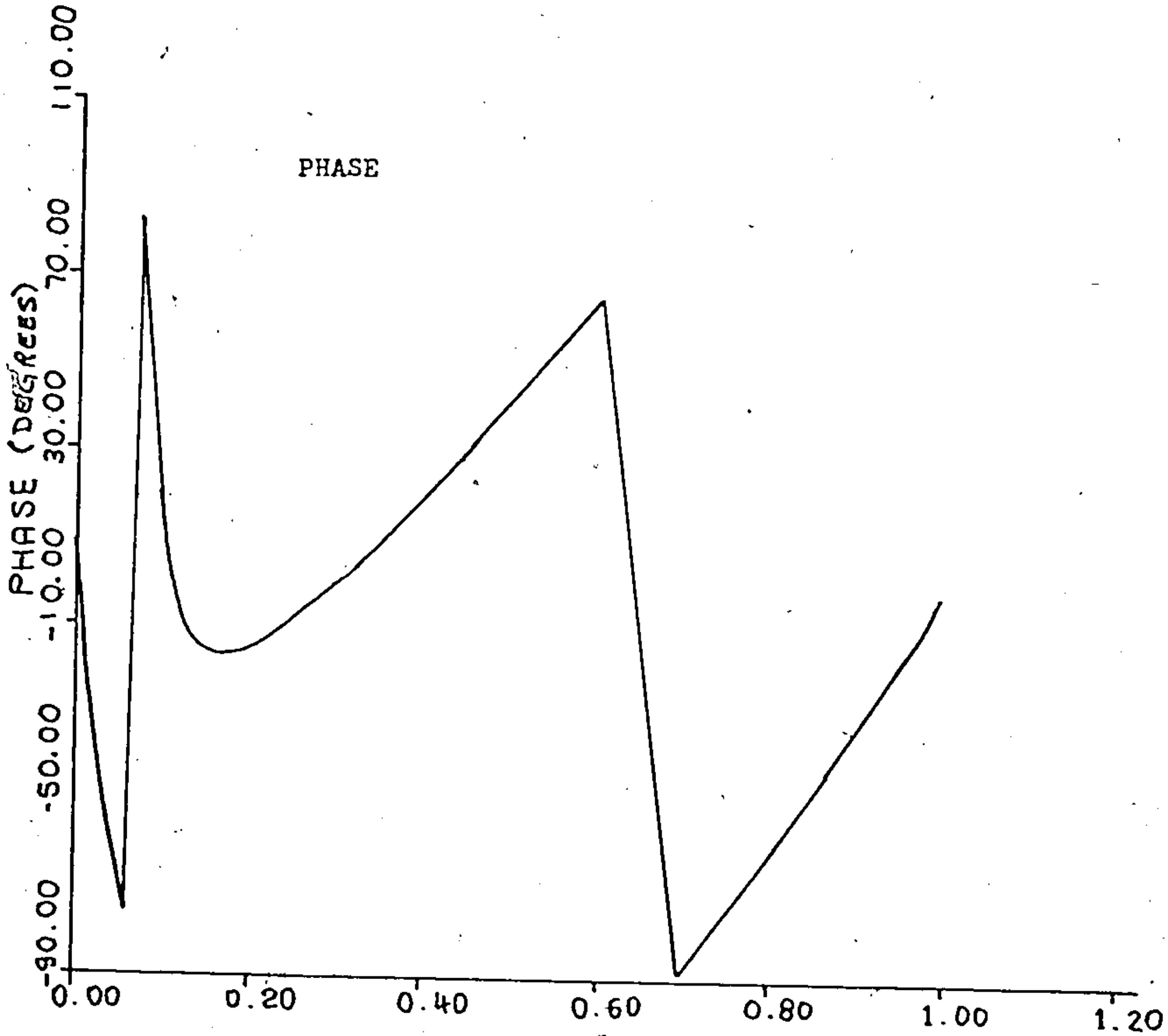
Example 1: Third order recursive digital filter

Q= 0.3659

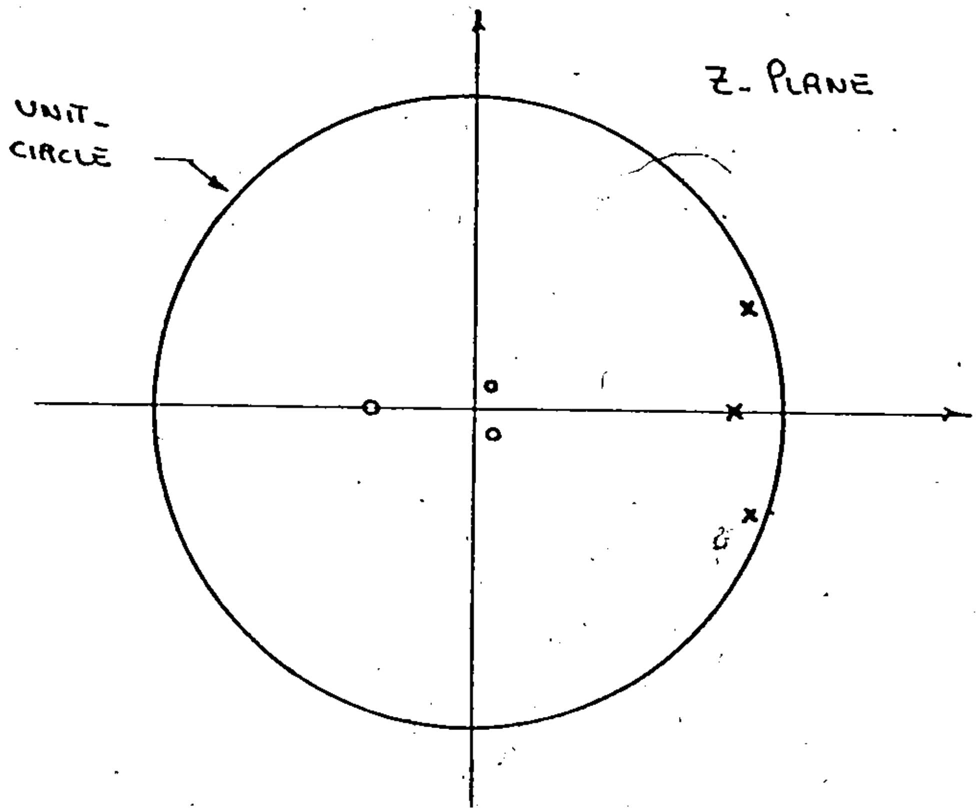
(10 iterations)



FREQUENCY IN FRACTIONS OF THE NYQUIST FREQUENCY



FREQUENCY IN FRACTIONS OF THE NYQUIST FREQUENCY



POLE-ZERO LOCATION OF EXAMPLE 1 (THIRD - ORDER)

Then,

$a_1=-1.86333$

$b_1=0.99689$

$c_1=-1.80196$

$d_1=0.87173$

$a_2=-1.20277$

$b_2=0.93355$

$c_2=-1.78574$

$d_2=0.87935$

$a_{z1}=-0.6666$

$b_{p1}=0.87316$

Example 2 : Design a high pass recursive digital filter having the following specifications

$w=0.0,0.495(0.05);$ $Y^d=0.0$

$w=0.5;$ $Y^d=0.5$

$w=0.505,1(0.05);$ $Y^d=1.0$

K=1,L=1 was considerd , and ψ_o was choosen to be

$$\psi_o=(0.4,0.4,0.4,0.4;0.4,0.4)^T$$

After 30 iterations the results of the filter were

$Q=0.5969$

$A^*=0.3016$

$$\text{Poles}\begin{pmatrix}0.1487 \pm j0.1357\\ 0.2506\end{pmatrix}$$

$$\text{Zeros}\begin{pmatrix} 0.60866 \pm & j0.78877 \\ 0.91841 & \end{pmatrix}$$

$a_1=-1.21732$

$b_1=0.99263$

$c_1=-0.29748$

$d_1=0.0405510$

$a_{z1}=0.91841$

$b_{p1}=0.25068$

If, now, an eighth order high-pass filter was designed taking K=3, L=2, and ψ_0 was taken as the final ψ of the previous case.

ψ_0 =(1.63,0.914,-0.465,0.740;(repeated 3 times,since K=3); 1.64,0.2534;(repeated 2 times,since L=2)).

After 17 iterations

$Q=0.33179$

$A^*=0.06659$

$a_1=-0.10853\times 10$

$a_2=a_3=a_1$

$b_1=0.99316$

$b_2=b_3=b_1$

$c_1=-0.13864$

$c_2=c_3=c_1$

$d_1=0.28580$

$d_2=d_3=d_1$

Poles 1= $(0.69321\times10^{-1} \pm j0.53009)$

=complex poles of the first cascade

=Poles 2 =Poles 3.

Zeros 1= $(0.54266 \pm j0.83587)$

=complex zeros of the first cascade

=Zeros 2 =Zeros 3.

$a_{z1}=0.91782$

$a_{z2}=a_{z1}$

$b_{p1}=0.51083$

$b_{p2}=b_{p1}$

After 30 iterations

$Q=0.091689$

$A^*=0.05905$

$a_1=-1.01448$

$b_1=0.86266$

$c_1=-0.14435$

$d_1=0.328583$

$a_2=-1.06575$

$b_2=0.92916$

$c_2=0.06223$

$d_2=0.46850$

$a_3=-1.03250$

$b_3=0.89096$

$c_3=0.01237$

$d_3=0.34901$

$a_{z1}=0.99286$

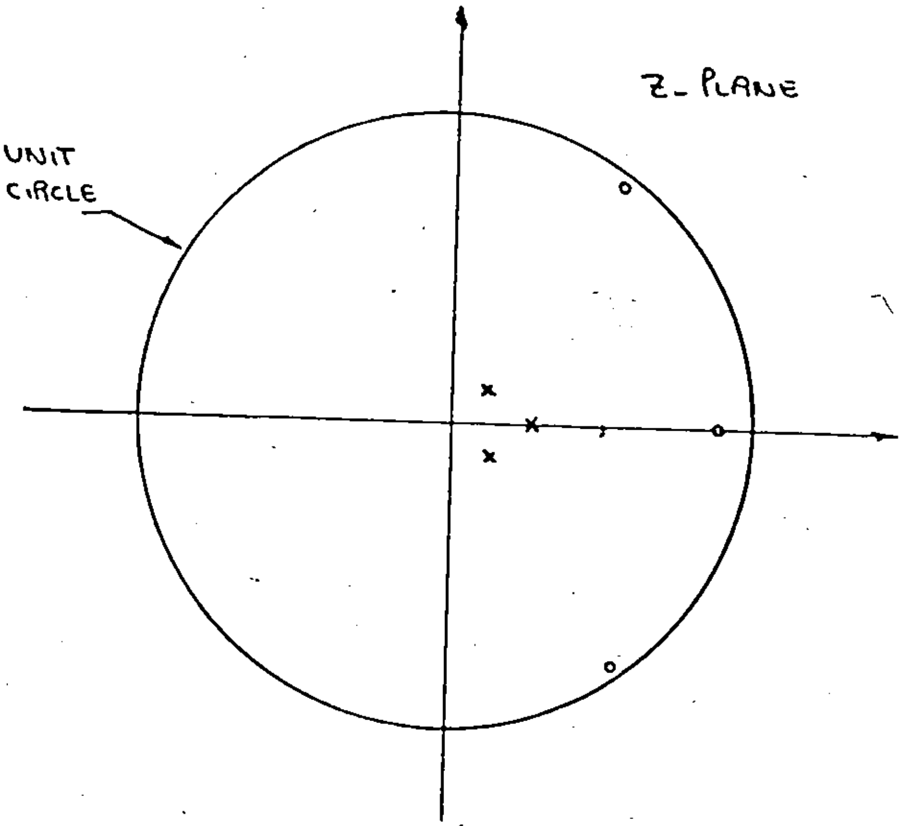
$b_{p1}=0.52533$

$a_{z2}=0.99761$

$b_{p2}=0.48769$

Poles
$(0.72175\times10^{-1} \pm j0.56866)$
$(-0.311163\times10^{-1} \pm j0.6837)$
$(-0.61889\times10^{-2} \pm j0.59074)$
(b_{p1}, and b_{p2})

Zeros
$(0.507240 \pm j0.778058)$
$(0.532874 \pm j0.803245)$
$(0.516253 \pm j0.790220)$
(a_{z1}, and a_{z2})

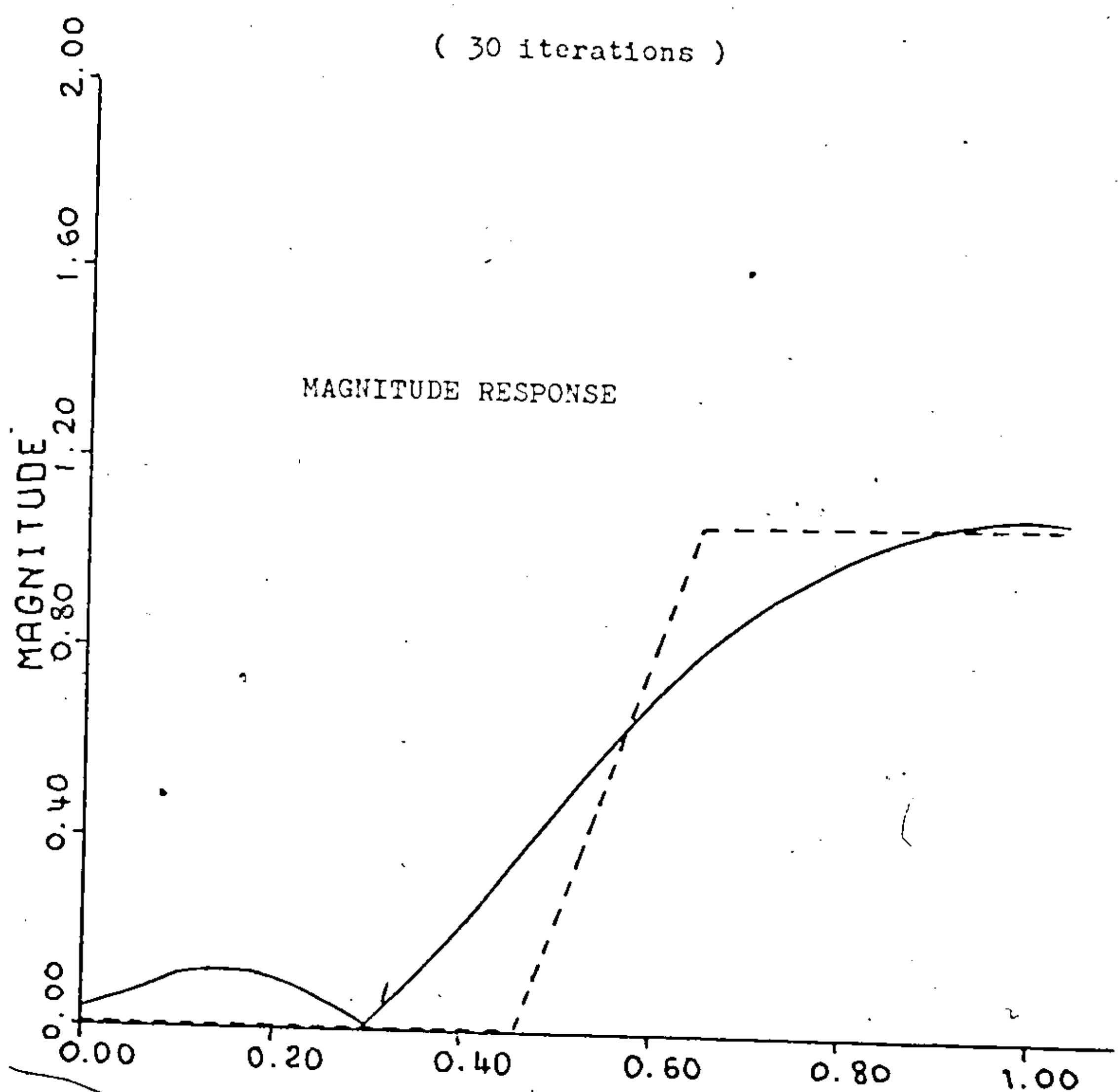


POLE-ZERO LOCATION OF EXAMPLE 2 (THIRD ORDER)

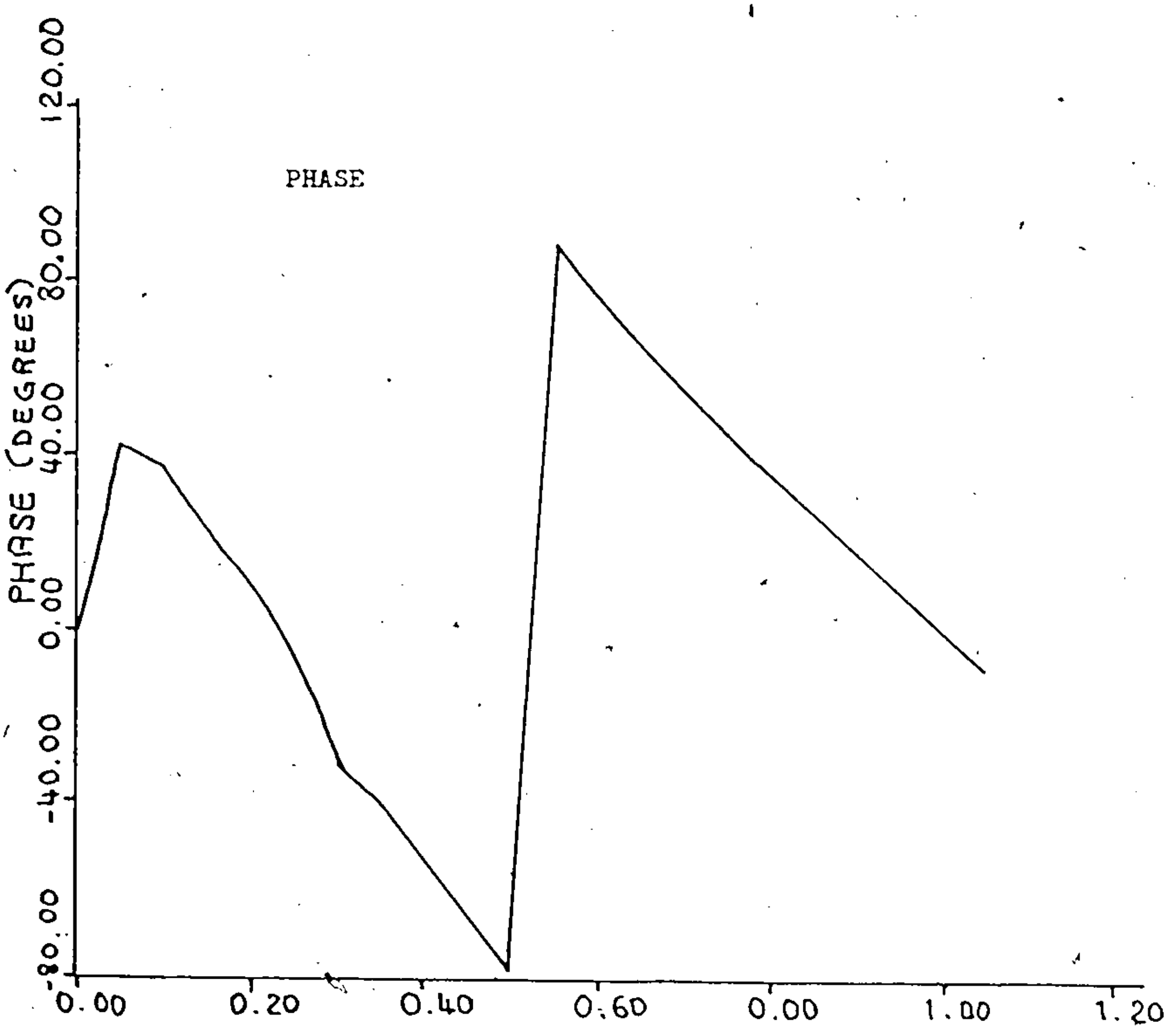
Example 2 : Third order digital high-pass filter

Q = 0.5969

(30 iterations)

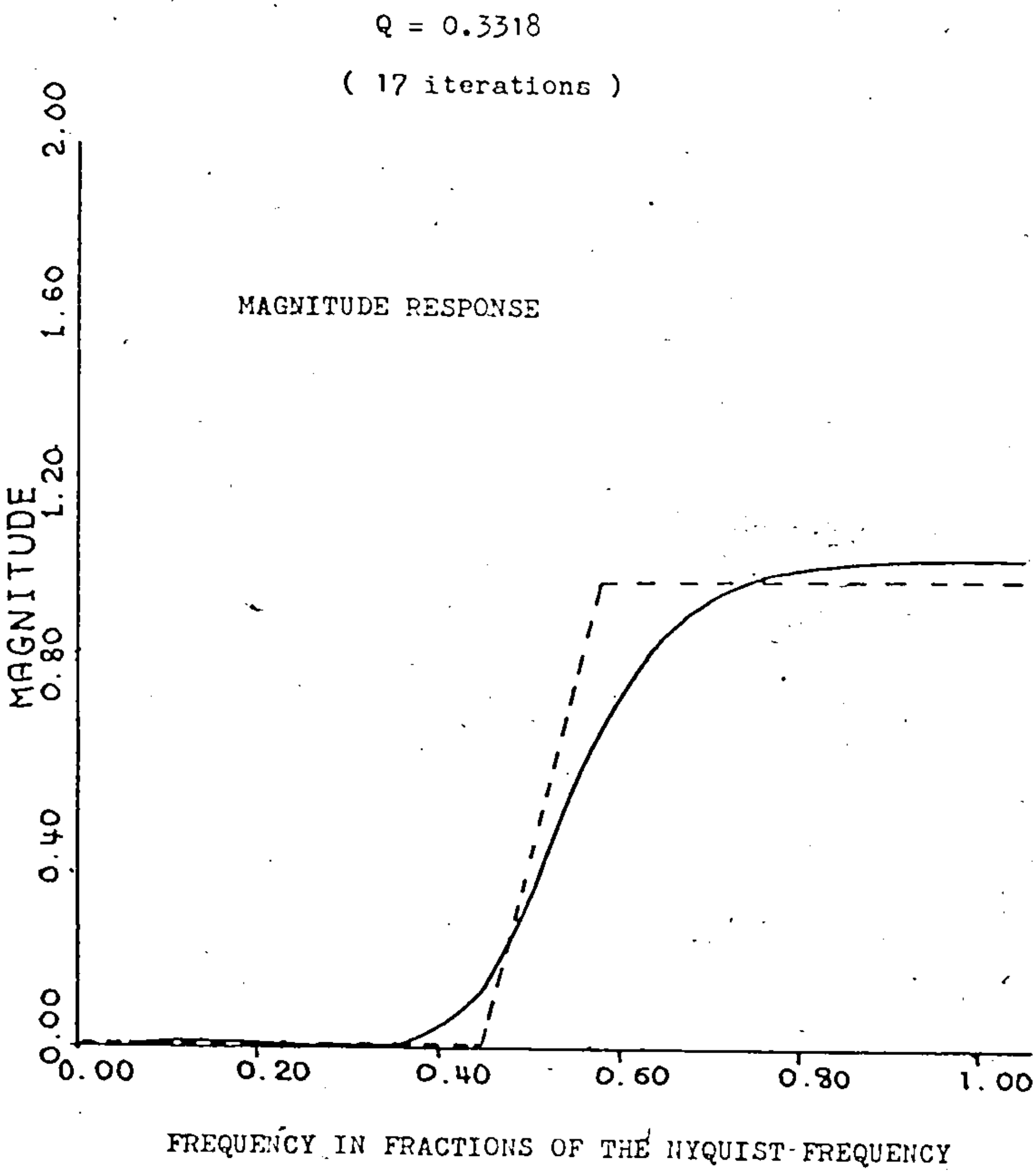


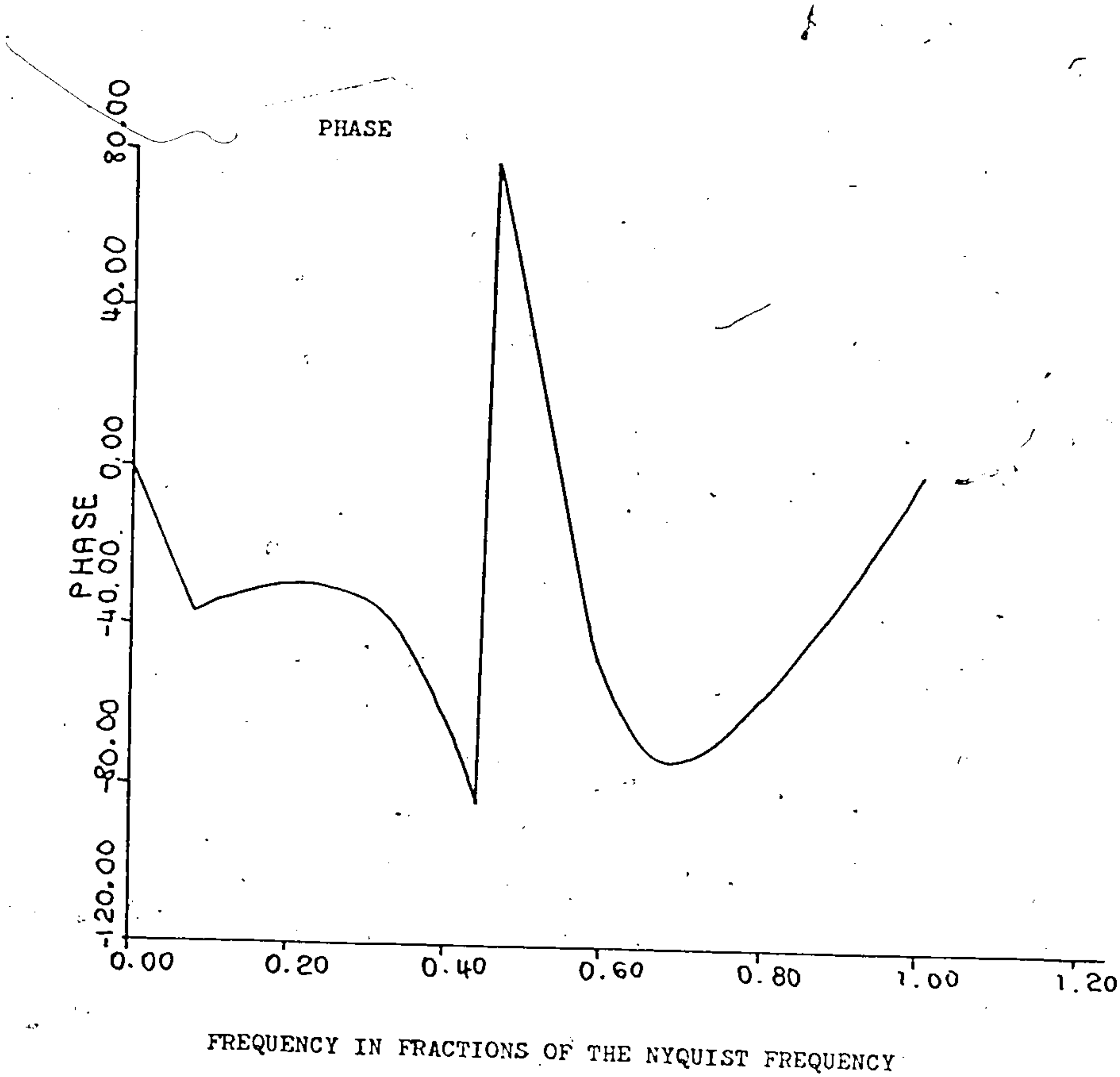
FREQUENCY IN FRACTIONS OF THE NYQUIST FREQUENCY



FRQUENCY IN FRACTIONS OF THE NYQUIST FRQUENCY

Example 2 : Eighth order digital high-pass filter

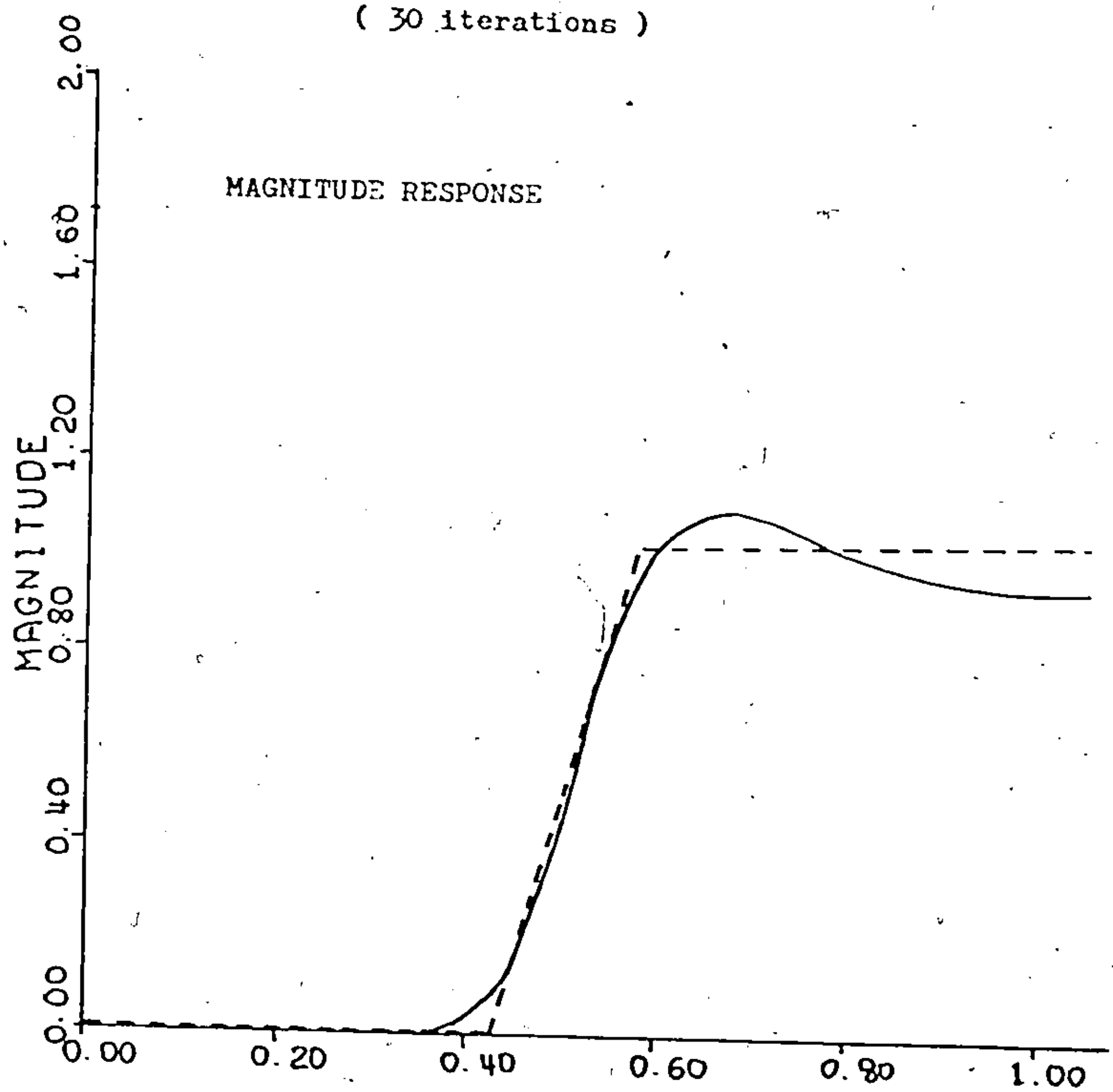



PHASE
PHASE
80.00
40.00
0.00
-40.00
-80.00
-120.00
0.00
0.20
0.40
0.60
0.80
1.00
1.20
FREQUENCY IN FRACTIONS OF THE NYQUIST FREQUENCY

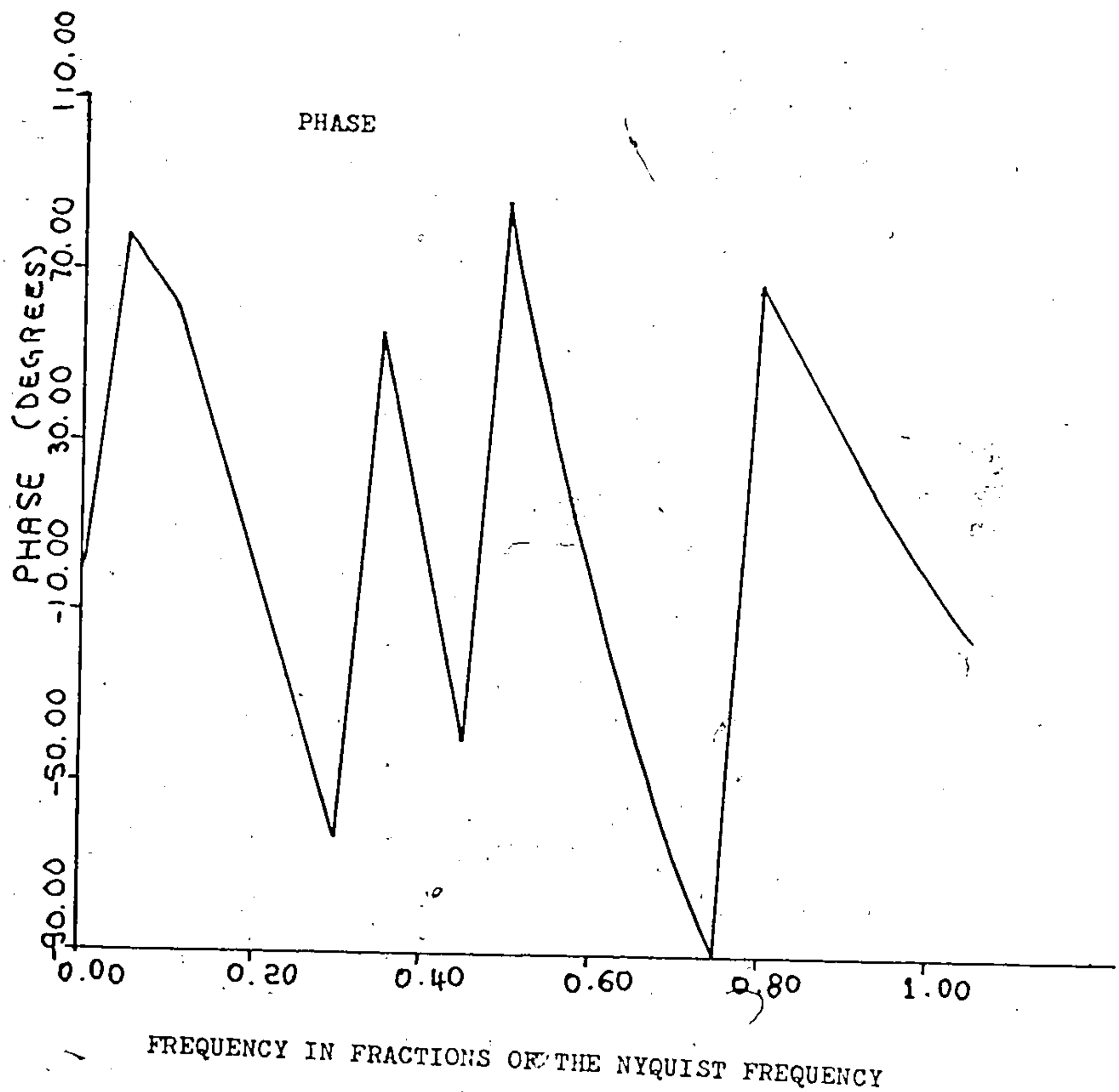
Example 2 : Eighth order digital high-pas filter.

Q = 0.0916898

(30 iterations)



FREQUENCY IN FRACTIONS OF THE NYQUIST FREQUENCY


PHASE
PHASE (DEGREES)
110.00
70.00
30.00
-10.00
-50.00
-90.00
0.00
0.20
0.40
0.60
0.80
1.00
FREQUENCY IN FRACTIONS OF THE NYQUIST FREQUENCY

Example 3 : Design a recursive digital band-pass filter having the following specifications

$w=0.0,0.32(0.08);$ $\quad Y^d=0.0$

$w=0.4;$ $\quad Y^d=0.5$

$w=0.402,0.508(0.02);$ $\quad Y^d=1.0$

$w=0.60;$ $\quad Y^d=0.5$

$w=0.604,1.0(0.04);$ $\quad Y^d=0.0$

Starting with $\Psi_o=(0.4,0.4,0.4,0.4;0.4,0.4)^T$, and taking K=L=1, after 30 iterations

$Q=1.3190$

$A^*=0.2412$

$a_1=-0.4483\times 10^{-3}$

$b_1=0.8809\times10^{-7}$

$c_1=0.0106365$

$d_1=0.74572$

$a_{z1}=0.93338$

$b_{p1}=0.49479$

Poles $(-0.531824\times 10^{-2}\pm\ j0.863534)$
$(\ b_{p1}\ )$

Zoros $(0.22415\times 10^{-3}\pm\ j0.194518\quad 10^{-3})$
$(\ a_{z1}\ )$

If , now an eighth order band-pass filter was designed taking $K=3$, $L=2$; and ψ_0 was taken as the final ψ of the previous case, i.e. ψ_0 =(0.017228,-0.71479,1.19244,1.5769; (repeated 3 times, since $K=3$); 1.93787,0.517595;(repeated 2 times , since $L=2$)). After 30 iterations

$$Q = 0.642177$$

$$A^* = 0.104296$$

$$a_1 = -0.169412 \times 10^{-3}$$

$$b_1 = 0.125339 \times 10^{-7}$$

$$c_1 = -0.61250 \times 10^{-1}$$

$$d_1 = 0.475182$$

$$a_2 = -0.16941 \times 10^{-3}$$

$$b_2 = b_1$$

$$c_2 = 0.250454$$

$$d_2 = 0.617215$$

$$a_3 = -0.16038 \times 10^{-3}$$

$$b_3 = 0.112358 \times 10^{-7}$$

$$c_3 = -0.21611$$

$$d_3 = 0.438632$$

$$a_{z1} = 0.94557$$

$$b_{p1} = 0.48471$$

$$a_{z2} = a_{z1}$$

$$b_{z2} = 0.48172$$

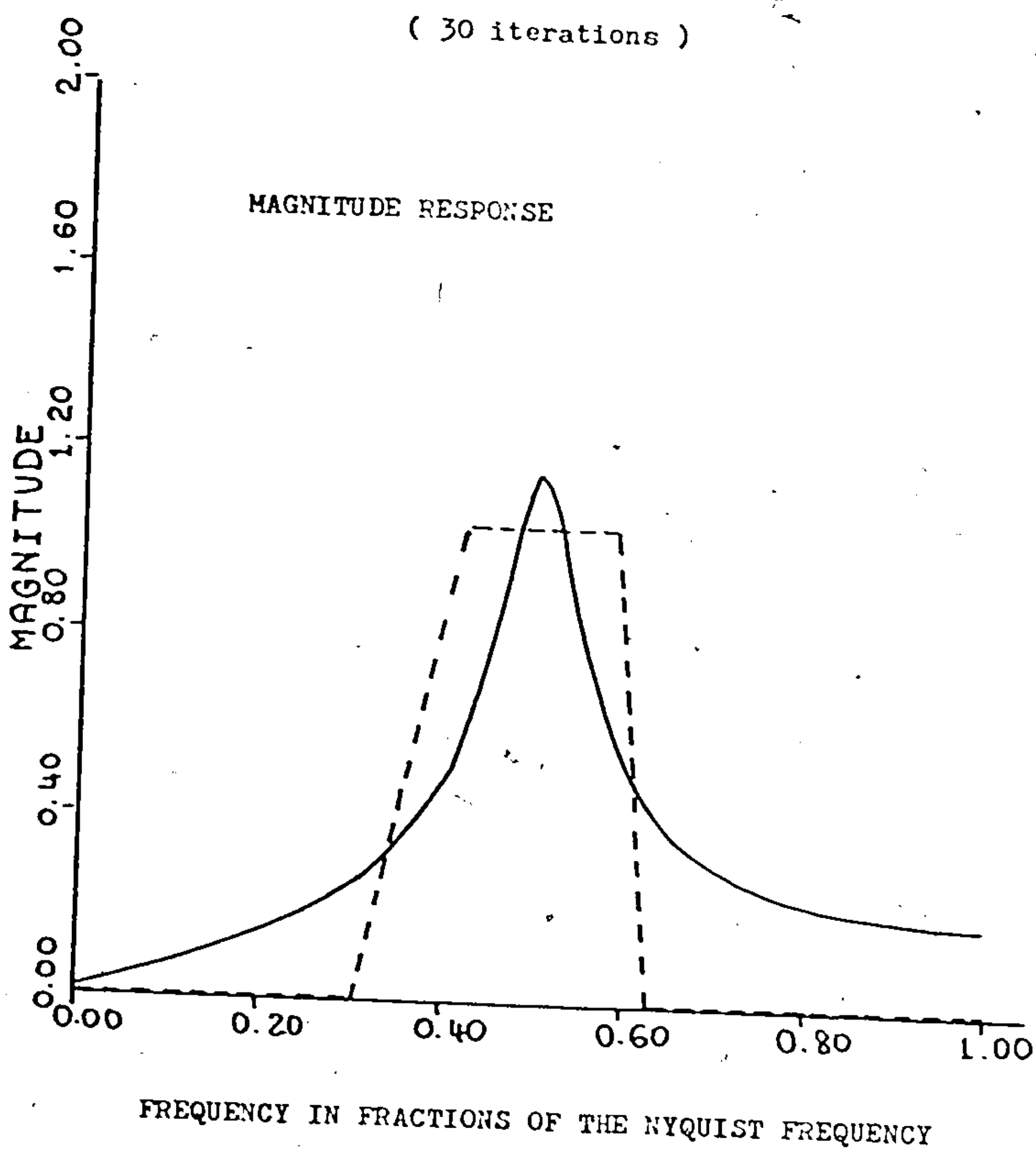
$$\text{Poles} \begin{array}{lll} (0.30625 \times 10^{-1} & \pm\, j0.68865 &) \\ (-0.1252 & \pm\, j0.77558 &) \\ (0.10805 & \pm\, j0.65342 &) \\ (b_{p1} \text{ and } b_{p2} & &) \end{array}$$

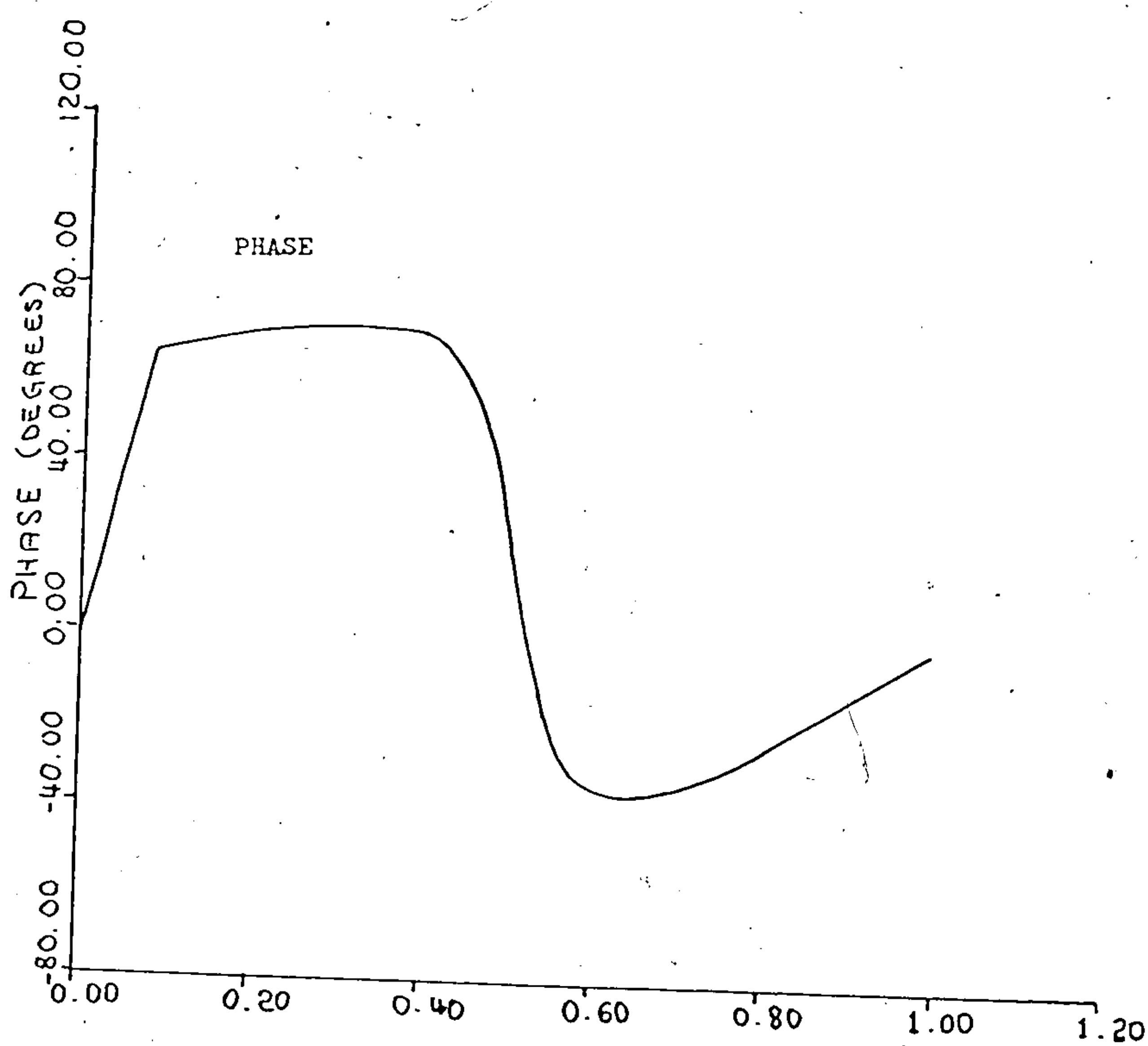
$$\text{Zeros} \begin{array}{lll} (0.847064 \times 10^{-4} & \pm\, j0.73203 & 10^{-4}) \\ (0.847064 \times 10^{-4} & \pm\, j0.73203 & 10^{-4}) \\ (0.801906 \times 10^{-4} & \pm\, j0.69320 & 10^{-4}) \\ (a_{z1} \text{ and } a_{z2} & &) \end{array}$$

Example 3 : Third order band-pass filter

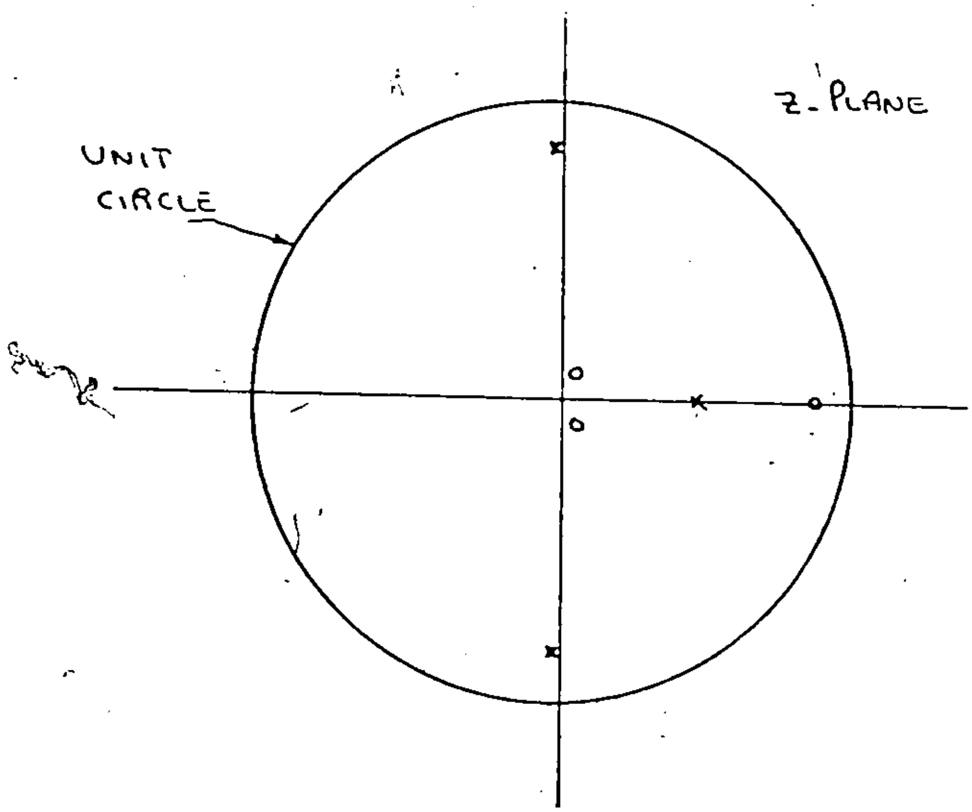
Q = 1.3198

(30 iterations)





FREQUENCY IN FRACTIONS OF THE NYQUIST FREQUENCY

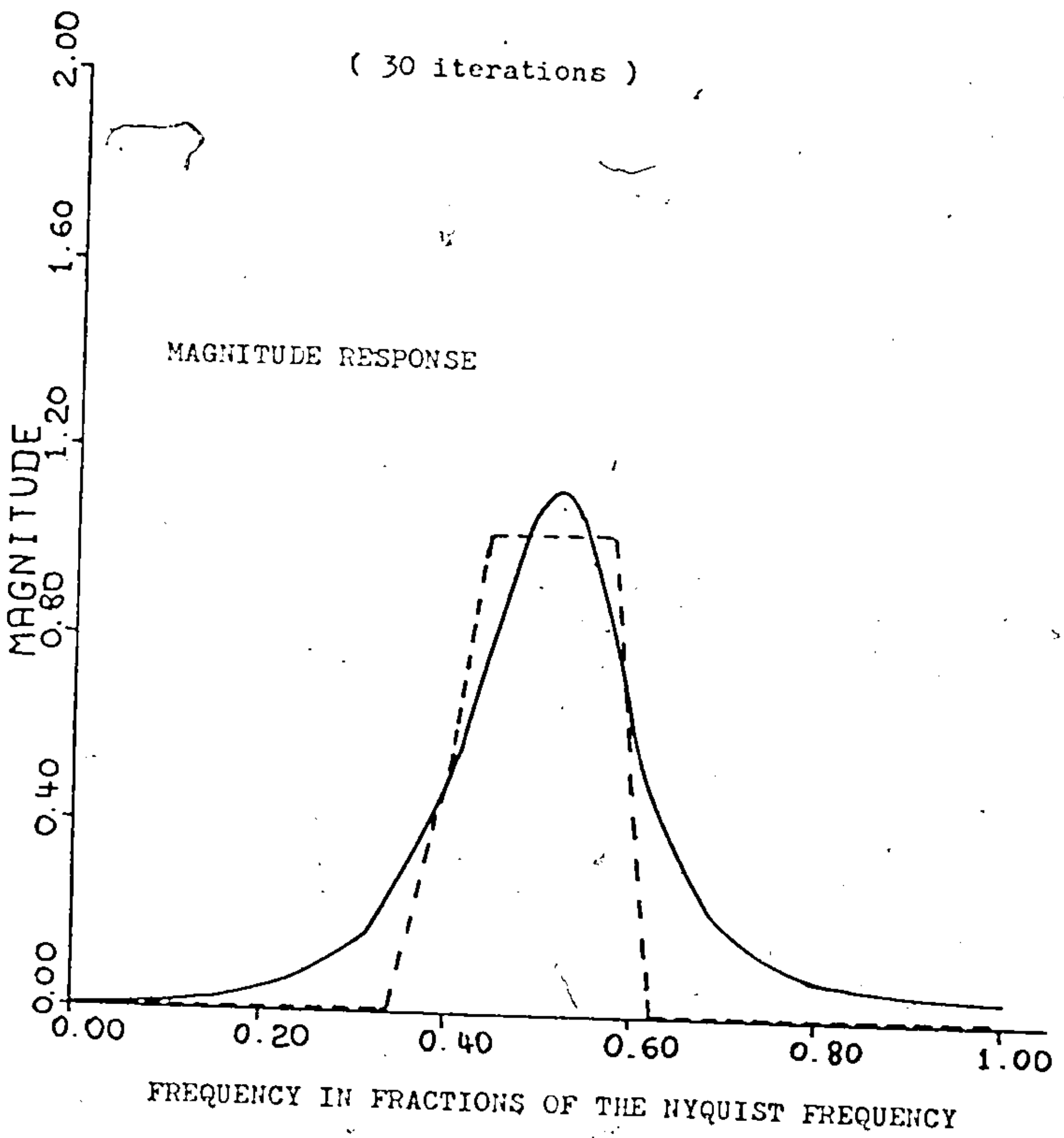


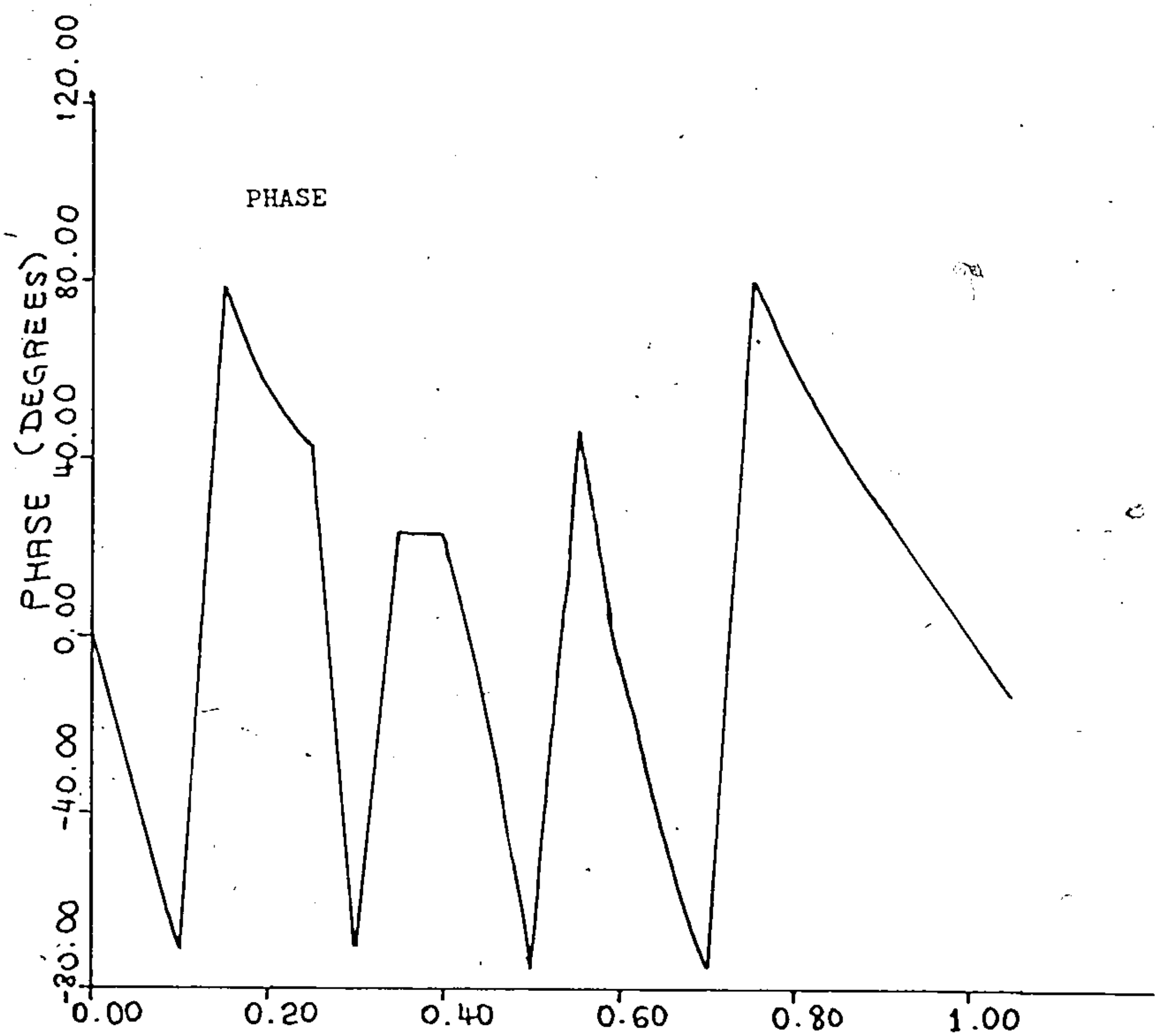
POLE-ZERO LOCATION OF THIRD ORDER BAND-PASS FILTER

Example 3 : eighth-order band-pass digital filter

Q =0.64178

(30 iterations)





FREQUENCY IN FRACTIONS OF THE NYQUIST FRQUENCY

3.10 A REVIEW OF STEIGLITZ'S METHOD[1]

A technique of minimizing a square-error criterion to obtain a required digital filter transfer function has been reported by Steiglitz[1].

This chapter introduces an improvement on Steiglitz's method of implementing constraints on the poles and zeros.

The method adopted in ref.(1) is as follows :

If Y(z) has a pole or a zero outside the unit circle , then inverting this pole or zero with respect to the unit circle does not affect the magnitude response , since the effect of inverting a pole at $|\alpha|e^{j\theta}$ with respect to the unit circle is a multiplication by the function

$$\frac{z - |\alpha|e^{j\theta}}{z - \frac{1}{|\alpha|}e^{j\theta}}$$

Now, $$z - |\alpha|e^{j\theta} = (1 - 2|\alpha|\cos(w\tau - \theta) + |\alpha|^2)^{\frac{1}{2}}$$

$$z - \frac{1}{|\alpha|}e^{j\theta} = (1 - 2\frac{1}{|\alpha|}\cos(w\tau - \theta) + \frac{1}{|\alpha|^2})^{\frac{1}{2}}$$

$$= \frac{1}{|\alpha|}(1 - 2|\alpha|\cos(w\tau - \theta) + |\alpha|^2)^{\frac{1}{2}}$$

This is effectively multiplying the magnitude of Y(z) by $|\alpha|$, since A is chosen optimally this does not affect the magnitude response.

The steps given in ref.(1) are as follows :

1) Start from a transfer function

$$Y(z,X)= A \prod_{k=1}^{K} \frac{1+a_k z^{-1}+b_k z^{-2}}{1+c_k z^{-1}+d_k z^{-2}}$$

where, $X=(a_k, b_k, c_k, d_k)^T$

2) Minimize the square error criterion given by eq.(3.8) .

3) At the optimum,(or a specified number of iterations), invert the poles or zeros lying outside the unit circle.

It has also been found that when starting anew from the result of step 3, further reduction in Q is possible.

The principal disadvantage of this procedure is the necessity of a search in the infinite z-plane .

It is possible to carry out this search in the unit circle where the poles and zeros are required to be located.

The method adopted in this chapter permits determination of a small value of Q in a relatively small number of iterations as compared with the method proposed by Steiglitz(1),irrespective of the initial pole-zero location.

In example 1, which was taken from ref.(1),Q reached a value of 0.3659 in 10 iterations using the method proposed in this chapter, while it took 93 iterations to reach a value of Q=1.2611 (for K=1), and a value of Q= 0.56731 in 155 iterations ,using Steiglitz's method.

CHAPTER IV

THE DESIGN OF ANALOG FILTERS USING MINIMIZATION TECHNIQUE

4.1 INTRODUCTION

The purpose of this chapter is to describe a method for choosing the coefficients of an analog filter to meet arbitrary specifications of magnitude response.

The method described does not design a filter for the whole infinite frequency range, but only a specified finite range of frequency.

The proposed method uses the optimization algorithm described by Fletcher and Powell to minimize a square-error criterion in the frequency domain.

4.2 THE FILTER FORM

To control the pole and zero locations of the analog filter, the following cascade form is chosen

$$G(s)= A \prod_{k=1}^{K} \frac{(s-\rho_k^2\, e^{j\phi_k})(s-\rho_k^2\, e^{j\phi_k})}{(s-p_k^2\, e^{j\theta'_k})(s-p_k^2\, e^{j\theta'_k})} \prod_{j=1}^{L} \frac{(s+\sigma_{zj}^2)}{(s+\sigma_{pj}^2)} \qquad (4.1)$$

which can be written in the form

$$G(s) = A \prod_{k=1}^{K} \frac{s^2 - (2\zeta_k^2.\cos\alpha_k)s + \zeta_k^4}{s^2 - (2p_k^2.\cos\theta_k')s + p_k^4} \prod_{j=1}^{L} \frac{(s + \sigma_{zj}^2)}{(s + \sigma_{pj}^2)} \quad (4.2)$$

4.3 THE CRITERION FUNCTION

Let the desired magnitude be described at a discrete set of frequencies $w_1, w_2, \ldots\ldots, w_M$, and let Y_i^d be the desired magnitude at the frequency w_i.

Choose a square-error criterion

$$Q(\psi_1) = \sum_{i=1}^{M} \left(|G(s_i, \psi_1)| - Y_i^d \right)^2 \quad (4.3)$$

where, $\psi_1 = (\zeta_k, \phi_k, p_k, \theta_k'; \sigma_{zj}, \sigma_{pj}; A)^T$, is the 4K+2L+1 vector of unknown coefficients, and

$$s_i = jw_i \quad (4.4)$$

4.4 ELIMINATION OF A AS AN UNKNOWN PARAMETER

To eliminate A from Q , define the 4K+2L dimensional parameter vector

$$\psi = (\zeta_k, \phi_k, p_k, \theta_k'; \sigma_{zj}, \sigma_{pj})^T \quad (4.5)$$

and write

$$G(s, A, \psi) = AH(s, \psi) \quad (4.6)$$

Then

$$Q(A,\psi) = \sum_{i=1}^{M}(|AH(S_i,\psi)| - Y_i^d)^2 \tag{4.7}$$

Differentiate (4.7) with respect to A and set the result to zero

$$\frac{\partial Q(A,\psi)}{\partial |A|} = 2\sum_{i=1}^{M}(|AH(s_i,\psi)| - Y_i^d)\,|H(s_i,\psi)| = 0.$$

Therefore

$$|A^*| = \frac{\sum_{i=1}^{M}|H(s_i,\psi)|\,Y_i^d}{\sum_{i=1}^{M}|H(s_i,\psi)|^2} \tag{4.8}$$

Since the sign of A^* does not affect the magnitude characteristic, it will be taken as positive.

The Fletcher-Powell method is used to minimize the new error-criterion

$$Q(\psi_i) = Q(A^*,\psi) \tag{4.9}$$

4.5 CALCULATION OF THE GRADIENT OF Q WITH RESPECT TO ψ

The Fletcher-Powell method requires the calculation of Q with respect to the parameter vector ψ.

Equations expressing the gradient of Q with respect to ψ are given in section 3.5 .

4.6 STABILITY CONSIDERATIONS

For the filter to be stable with minimum phase constraint the poles and zeros should lie in the left-half of the s-plane.

Hence, $\frac{\pi}{2} \leq \phi_k \leq \pi$

and using variable substitution

$$\phi_k = \frac{\pi}{2} + \frac{\pi}{2} \sin^2 \alpha_k \tag{4.10}$$

also $\frac{\pi}{2} \leq \theta_k' \leq \pi$

using variable substitution

$$\theta_k' = \frac{\pi}{2} + \frac{\pi}{2} \sin^2 \theta_k \tag{4.11}$$

where θ_k and α_k are the substitution variables.

Substituting (4.10) and (4.11) we get

$$G(s) = A \prod_{k=1}^{K} \frac{s^2 + 2\rho_k^2 . \sin(\frac{\pi}{2} \sin^2 \alpha_k) . s + \rho_k^4}{s^2 + 2p_k^2 . \sin(\frac{\pi}{2} \sin^2 \theta_k) . s + p_k^4} \prod_{j=1}^{L} \frac{s + \sigma_{zj}^2}{s + \sigma_{pj}^2} \tag{4.12}$$

4.7 CALCULATION OF $\dfrac{\partial |H(s_i,\psi)|}{\partial \psi}$

Since,

$$\frac{\partial |H(s_i,\psi)|}{\partial \psi} = \frac{1}{|H(s_i,\psi)|} \cdot \mathrm{Re}\left(\overline{H(s_i,\psi)} \cdot \frac{\partial H(s_i,\psi)}{\partial \psi} \right)$$

Therefore

$$\frac{\partial |H(s_i,\psi)|}{\partial \rho_k} = |H_i| . \mathrm{Re}\left(\frac{4\rho_k . \sin(\frac{\pi}{2}\sin^2\alpha_k) . s_i + 4\rho_k^3}{s_i^2 + 2\rho_k^2 . \sin(\frac{\pi}{2}\sin^2\alpha_k) . s_i + \rho_k^4} \right)$$

$$\frac{\partial |H_i|}{\partial \alpha_k} = |H_i| . \mathrm{Re}\left(\frac{2\rho_k^2 . \cos(\frac{\pi}{2}\sin^2\alpha_k) . \frac{\pi}{2} . 2\sin\alpha_k . \cos\alpha_k . s_i}{s_i^2 + 2\rho_k^2 . \sin(\frac{\pi}{2}\sin^2\alpha_k) . s_i + \rho_k^4} \right)$$

$$\frac{\partial |H_i|}{\partial p_k} = -|H_i| . \mathrm{Re}\left(\frac{4p_k . \sin(\frac{\pi}{2}\sin^2\theta_k) . s_i + 4p_k^3}{s_i^2 + 2p_k^2 . \sin(\frac{\pi}{2}\sin^2\theta_k) . s_i + p_k^4} \right)$$

$$\frac{\partial |H_i|}{\partial \theta_k} = -|H_i| . \mathrm{Re}\left(\frac{2p_k^2 . \cos(\frac{\pi}{2}\sin^2\theta_k) . \pi . \sin\theta_k . \cos\theta_k . s_i}{s_i^2 + 2p_k^2 . \sin(\frac{\pi}{2}\sin^2\theta_k) . s_i + p_k^4} \right)$$

$$\frac{\partial |H_i|}{\partial \sigma_{zj}} = |H_i| . \operatorname{Re}\left(\frac{2\sigma_{zj}}{s + \sigma_{zj}^2}\right)$$

$$\frac{\partial |H_i|}{\partial \sigma_{pj}} = -|H_i| . \operatorname{Re}\left(\frac{2\sigma_{pj}}{s + \sigma_{pj}^2}\right)$$

4.8 SUMMARY OF THE SUBROUTINE FUNCT(N,X,Q,DQ)

This subroutine evaluates Q and grad.Q , and is needed when using the Fletcher-Powell algorithm.

1) Calculate H_i $\quad i=1,2,......,M$

2) Calculate

$$A^* = \frac{\sum_{i=1}^{M} |H_i| . Y_i^d}{\sum_{i=1}^{M} |H_i|^2}$$

3) Calculate $E_i = A^* |H_i| - Y_i^d$ $\quad i=1,2,.....,M$

4) Calculate

$$Q = \sum_{i=1}^{M} E_i^2$$

5) Calculate

$$\frac{\partial |H_i|}{\partial \psi_n} \qquad n=1,2,.....,4K+2L$$

(Formulas are given in previous section)

6) Calculate

$$\frac{\partial Q}{\partial \psi_m} = 2 A^* \sum_{i=1}^{M} E_i \frac{\partial |H_i|}{\partial \psi_m} \qquad n = 1, 2, \ldots, 4K+2L$$

This subroutine was written in double precision complex arithmetic in Fortran IV , and was used in conjunction with the subroutine DFMFP (double-precision Fletcher-Powell algorithm) sublied by IBM.

4.9 EXAMPLE

Design an equalizer , whose specified gain response in the frequency range $1000/2 \le f \le 2000/2$, is given by

$$Gain = -8 + (9/1000) * w$$

The order of the filter is to be three(K=1, L=1).

$$\Psi = (\rho_k, \alpha_k, p_k, \theta_k; \sigma_{zj}, \sigma_{pj})^T$$

was choosen to be

$$\Psi_0 = (0.4, 0.4, 0.4, 0.4; 0.4, 0.4)^T \quad .$$

After 40 iterations

$$Q = 0.0144867$$

$$A^* = 31.4623$$

Poles $(-2495.37 \pm j65.5780)$
$(-73.3031 \qquad)$

Zeros $(-42.43711 \pm j87.60335)$
(-14.43188)

If we assume the filter to have the following form :

$$G(s) = A^{*} \frac{s^2 + a_1 s + b_1}{s^2 + c_1 s + d_1} \frac{s + \sigma_{z1}^2}{s + \sigma_{p1}^2}$$

then,

$a_1 = 84.8723$

$b_1 = 76.9235 \; 10^4$

$c_1 = 49.90737 \; 10^2$

$d_1 = 6.231164 \; 10^6$

$\sigma_{z1}^2 = 14.43188$

$\sigma_{p1}^2 = 73.30317$

CHAPTER V

DESIGN OF ANALOG FILTERS USING THE BILINEAR TRANSFORM

5.1 INTRODUCTION

In the design of analog filters , using optimization techniques, one encounters the problem of the infinite frequency axis. For a design specified over the complete frequency range , the frequency axis has to be divided into an infinite number of samples , specifying the magnitude at each frequency sample.

An interesting solution to this problem is to use the bilinear transformation to map the infinite frequency axis into a finite contour (circle) in the z-plane.

The problem is now reduced to one of choosing the analogue filter specifications (including the gain of the filter required at infinity) and transforming these specifications to the z-plane.

A solution is now sought for a digital filter with these transformed specifications and the solution transformed back to the s-plane.

The bilinear transform , which maps the jw axis in the s-plane onto the unit circle in the z-plane, is given by

$$s = \frac{z-1}{z+1} \tag{5.1}$$

For every, $s=jw_a$ there is a corresponding $z=e^{jw_dT}$ (5.2)

where , w_a= analog frequency

w_d= digital frequency

Therefore,

$$jw_a = \frac{e^{jw_dT} - 1}{e^{jw_dT} + 1}$$

From which

$$w_{ai} = \tan(w_{di}T/2) = \tan(\pi w_i/2) \qquad (5.3)$$

where w_i is given in fractions of the Nyquist rate.

5.2 FREQUENCY SCALING OF THE ANALOG FILTER

Suppose w_i=1.0 , then

$w_a = \tan(\pi/2) = \infty$ rad./sec. = ∞ Hz.

Now, consider w_i=0.9, then

$w_a = \tan(\pi * 0.9/2) = \tan 81^\circ = 6.314$ rad./sec. $\doteq$ 1 Hz.

That is, frequencies ranging from 1 Hz. to infinite Hz. on the jw axis of the s-plane , are all located in a period of 0.1 the Nyquist period. The design of a digital filter with desired specifications in such a small range is impracticable.

To remedy this effect we specify the desired filter at certain samples , then linearly scale the frequency axis such that the maximum frequency sample is scaled (say) to 1 rad./sec.. Thus the filter specifications when transferred into the z-domain are all located between 0.0, and 0.5 the Nyquist rate.

Samples between zero Hz., and the minimum frequency, and between the maximum frequency and infinite Hz, are specified as desired on the digital axis.

After the filter is designed in the z-domain, the solution is transformed to the s-domain and rescaled to the desired specifications.

The method of frequency scaling is shown with the following example.

Consider a band-pass filter having the transfer function

$$Y(s) = \frac{s}{s^2 + as + b} \tag{5.4}$$

This filter has poles at

$$s = \frac{-a}{2} \pm j\sqrt{b - \frac{a^2}{2}} \tag{5.5}$$

Its center-frequency

$$w_c = \sqrt{b} \tag{5.6}$$

and band-width

$$BW = a \tag{5.7}$$

Now, if we define the s'-plane, where $s' = Ks$, then the transformation of the singularities in the s -plane to the

s'-plane is

$$s' = Ks = -\frac{(Ka)}{2} \pm j\sqrt{(K^2b) - \frac{Ka}{2}^2}$$

Therefore, if s is replaced by s' in Eq.(5.4), and 'a' by Ka, and 'b' by K^2b, then it is obvious from equations (5.6), and (5.7) that w_c is shifted K times, and that the band-width is multiplied by K.

5.3 PROCEDURE

1) Define the required analog filter on the frequency axis at a finite number of samples.

2) Divide the prescribed frequencies by a scale factor, such that the maximum frequency specified is 1 rad./sec.

3) Using eq.(5.3),

$$w_i = \frac{2}{\pi}\tan^{-1}(w_{ai})$$

and the bilinear transformation is introduced to transform the analog frequency axis to the digital frequency axis.

4) Frequencies between the maximum frequency specified and infinite Hz are transferred between 0.5 and 1.0 the Nyquist rate on the digital frequency axis. This range is now divided into a finite number of frequencies, specifying at each frequency the desired amplitude at the higher ranges of frequencies in the s-domain.

5) Design the digital filter to meet the required magnitude response using the outlined optimization procedure.

6) Replace z by $\frac{1+s}{1-s}$ in the digital filter designed.

If the digital filter designed had the form

$$Y(z) = A\prod_{k=1}^{K}\frac{z^2 + a_k z + b_k}{z^2 + c_k z + d_k}\prod_{j=1}^{L}\frac{z - a_j}{z - b_j}$$

Then,

$$Y(s) = A'\prod_{k=1}^{K}\frac{s^2 + a_k' s + b_k'}{s^2 + c_k' s + d_k'}\prod_{j=1}^{L}\frac{s + a_j'}{s + b_j'}$$

where ,

$$A' = A\prod_{k=1}^{K}\frac{1 - a_k + b_k}{1 - c_k + d_k}\prod_{j=1}^{L}\frac{1 + a_j}{1 + b_j}$$

$$= \frac{\sum_{i=1}^{M}|H(s_i,\psi)| \cdot Y_i^d}{\sum_{i=1}^{M}|H(s_i,\psi)|^2} \qquad \text{(Chapter IV)}$$

$$a_k' = \frac{2 - 2b_k}{1 - a_k + b_k}$$

$$b_k' = \frac{1 + a_k + b_k}{1 - a_k + b_k}$$

$$c_k' = \frac{2 - 2d_k}{1 - c_k + d_k}$$

$$d_k' = \frac{1 + c_k + d_k}{1 - c_k + d_k}$$

$$a_j' = \frac{a_j - 1}{a_j + 1}$$

$$b_j' = \frac{b_j - 1}{b_j + 1}$$

7) If the frequency division scale factor is λ, then the desired analog filter should have the parameters

$$a_k' = \lambda a_k'$$

$$b_k' = \lambda^2 b_k'$$

$$c_k' = \lambda c_k'$$

$$d_k' = \lambda^2 d_k'$$

$$a_j' = \lambda a_j'$$

$$b_j' = \lambda b_j'$$

5.4 EXAMPLES

Example 1: Design a high-pass filter with maximum magnitude 1.0, and a value of 0.5 at 10 KHz.

In this example we shall alter the previous procedure presented, to show the method of scaling.

1- We first design a recursive digital high-pass filter having cut-off at 0.5 the Nyquist period.

The filter specifications are choosen to be

$$w = 0.00, 0.495(0.05); \qquad Y^d = 0.0$$
$$w = 0.5; \qquad Y^d = 0.5$$
$$w = 0.505, 1.0(0.05); \qquad Y^d = 1.0$$

ψ_0 is choosen to be $(0.4,0.4,0.4,0.4;0.4,0.4)^T$.
For K=L=1, and after 10 iterations the results are

$Q=0.5968$

$A^*=0.30021$

$a_1=-1.23881$

$b_1=0.998895$

$c_1=-0.30137$

$d_1=0.0406117$

$a_{j1}=0.89988$

$b_{j1}=0.24049$

Poles $(0.15068 \pm j0.13381)$
(0.24049)

Zeros $(0.61940 \pm j0.78436)$
(0.89988)

2- The filter is then transferred to the s-plane , and we get the following results

$A = 1.12269$

$a_1 = 0.68598 \times 10^{-3}$

$b_1 = 0.23476$

$c_1 = 1.42981$

$d_1 = 0.55086$

$a_{j1} = 0.5269 \times 10^{-1}$

$b_{j1} = 0.61226$

Poles $(-0.71490 \pm j0.48452)$
$(-b_{j1})$

Zeros $(-0.34299 \times 10^{-3} \pm j0.199425)$
$(-a_{j1})$

3- The scale factor is obtained as follows

$$w_a = \tan(w\pi/2) = \tan(\pi/4) = 1.0$$

$$f_a = 1.0/2\pi$$

$$\lambda = 10^4 \times 2\pi$$

4- The desired filter parameters are

$$a_1' = \lambda a_1$$

$$b_1' = \lambda^2 b_1$$

$$c_1' = \lambda c_1$$

$$d_1' = \lambda^2 d_1$$

$$a_{j1}' = \lambda a_{j1}$$

$$b_{j1}' = \lambda b_{j1}$$

The poles and zeros are each multiplied by λ.

Example 2 : Design a low-pass filter with cut-off at 10 Krad/sec.

According to the procedure given in sec.5.3

1- The filter was specified in the analog frequency domain as follows

$w_a = 0.0, 9500\ (950);$ $\quad Y^d = 1.0$

$w_a = 10^4\ ;$ $\quad Y^d = 0.707$

$w_a = 10^4, 15000\ (990);$ $\quad Y^d = 0.0$

2- The scale factor $\lambda = 1.0/15{,}000$

3- The filter specifications from 0.0 to 0.5, on the digital frequency axis are obtained as follows

$$w_i = \frac{2}{\pi} \tan^{-1}(w_{ai} \cdot \lambda)$$

4- The filter specifications from 15000 rad/sec.(0.5 the Nyquist rate on the digital frequency axis), and infinite rad/sec (the Nyquist rate) , are specified as follows

$$w_i = 0.5, 1.0(0.05); \qquad Y^d = 0.0$$

5-After designing the digital filter, starting with

$$\psi_0 = (0.4, 0.4, 0.4, 0.4; 0.4, 0.4)^T ;$$

and taking K=L=1 , we transform the digital filter to the s-domain, and scale the frequency axis as in sec.5.3 we obtain the required analog filter.

After 10 iterations the analog filer parameters were

$Q = 1.07466$

$A^* = 0.052539$

$a_1 = 0.318772 \times 10^5$

$b_1 = 0.254039 \times 10^9$

$c_1 = 0.97429 \times 10^4$

$d_1 = 0.67080 \times 10^8$

$a_{j1}=0.36211\times10^5$

$b_{j1}=0.73302\times10^4$

Poles $(-0.48714\times10^4 \pm j0.62515)$
$(-b_{j1})$

Zeros $(-0.159386\times10^5 \pm j0.65840\times10^4)$
$(-a_{j1})$

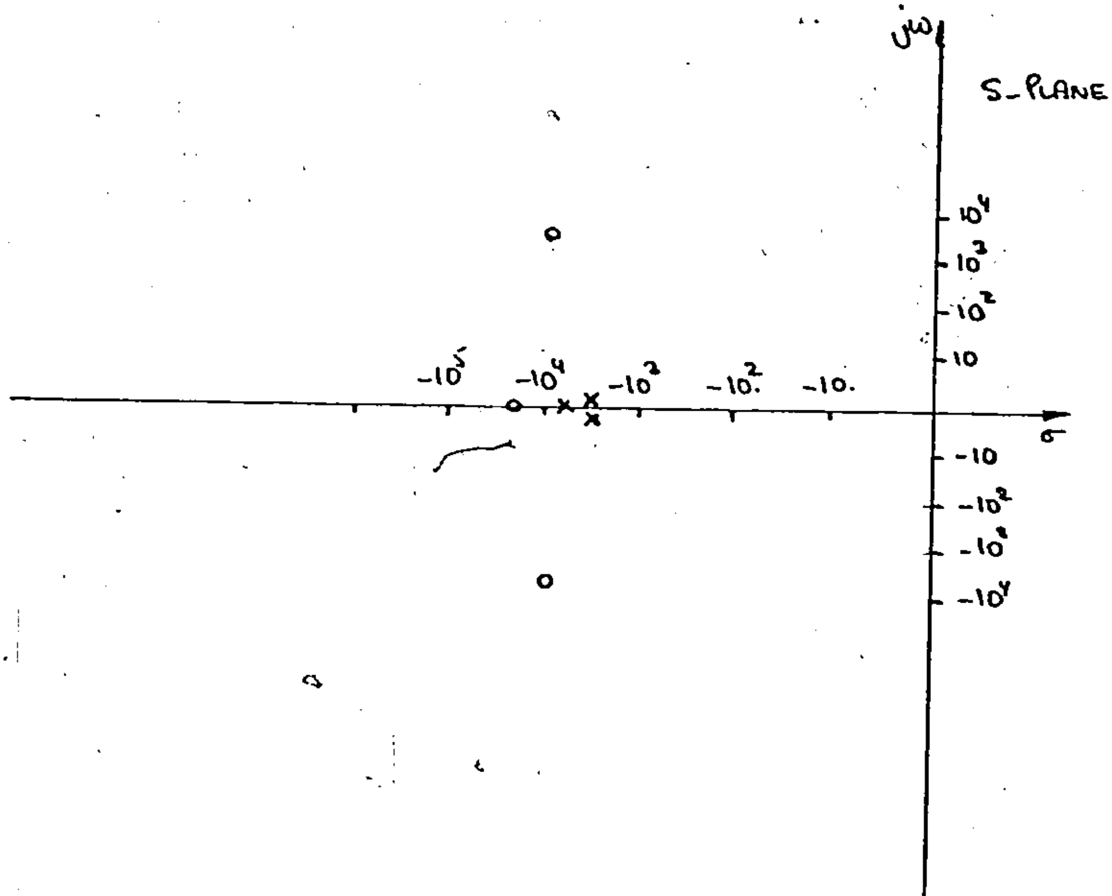


Fig. Pole-zero locatioh of the third order low-pass filter

After 30 iterations

$Q = 0.49738$

$A = 0.027621$

$a_1 = 0.31860 \times 10^5$

$b_1 = 0.25376 \times 10^9$

$c_1 = 0.37639 \times 10^4$

$d_1 = 0.58154 \times 10^8$

$a_{j1} = 0.25196 \times 10^5$

$b_{j1} = 0.26698 \times 10^4$

Poles $(-0.188197 \times 10^4 \pm j0.456155 \times 10)$
$(-b_{j1})$

Zeros $(-0.1593000 \times 10^5 \pm j0.739004 \times 10^4)$
$(-a_{j1})$

For an eighth-order filter K=3, L=2 was chosen, and ψ_o was chosen from the final conditions of the previous design.

ψ_o =(0.687x10^{-2} , 0.428 , 0.876 , 0.843,(repeated 3 times since K=3); -0.399 , 0.379, (repeated 2 times, since L=2)).

After 17 iterations the results of the analog filter were

$Q=0.22631$

$A=0.19915\times10^{-3}$

$a_1=0.3187\times10^{5}$

$a_2=a_3=a_1$

$b_1=0.25408\times10^{9}$

$b_2=b_3=b_1$

$c_1=0.70117\times10^{4}$

$c_2=c_3=c_1$

$d_1=0.78435\times10^{8}$

$d_2=d_3=d_1$

Poles 1 $(-0.35059\times10^{4}\pm j0.120843\times10^{-2})$

=complex poles of one cascade

=Poles 2 =Poles 3

Zeros 1 $(-0.159399\times10^{5}\pm j0.813286\ 10^{4})$

=complex zeros of one cascade

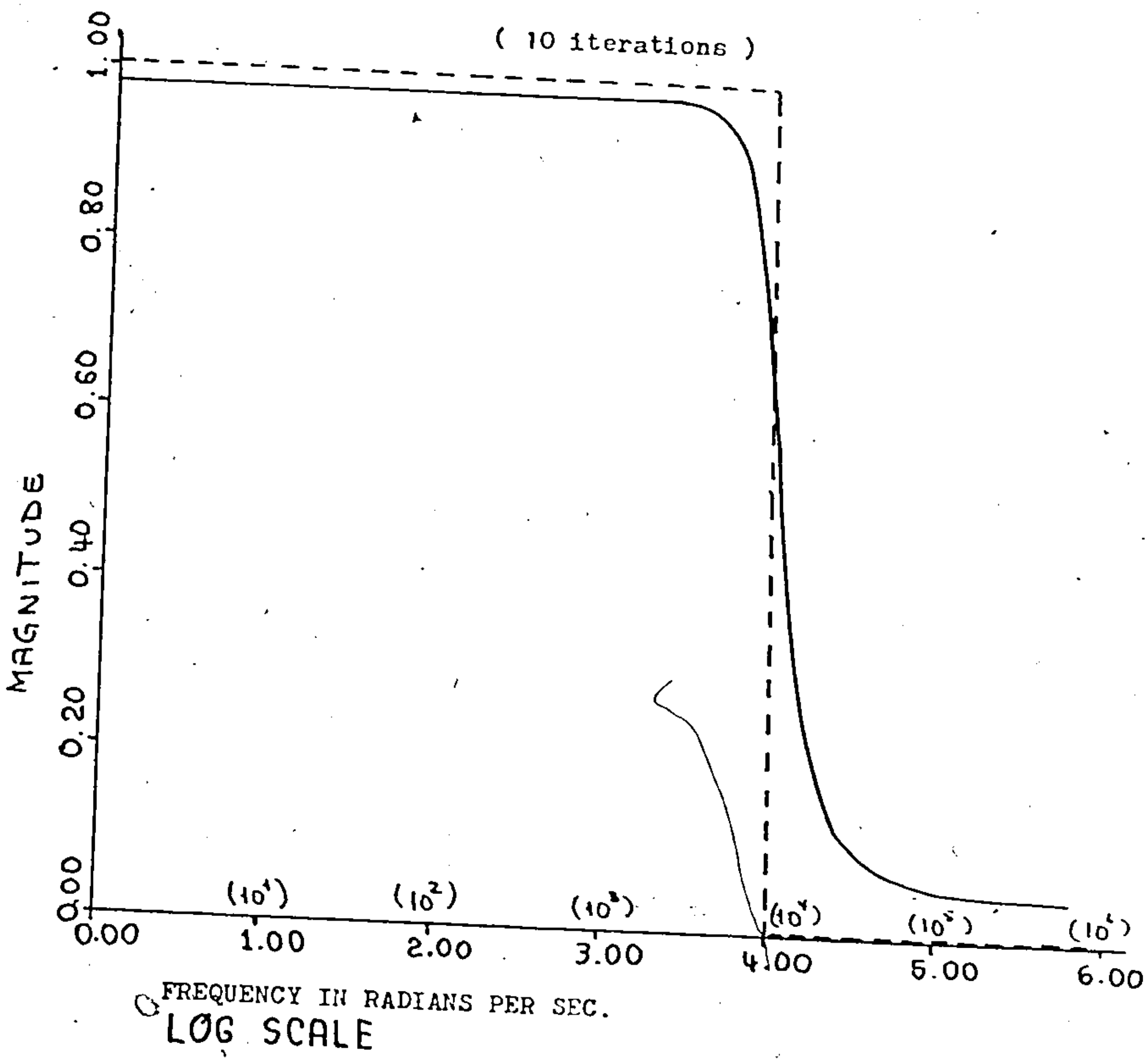
=Zeros 2 =Zeros 3

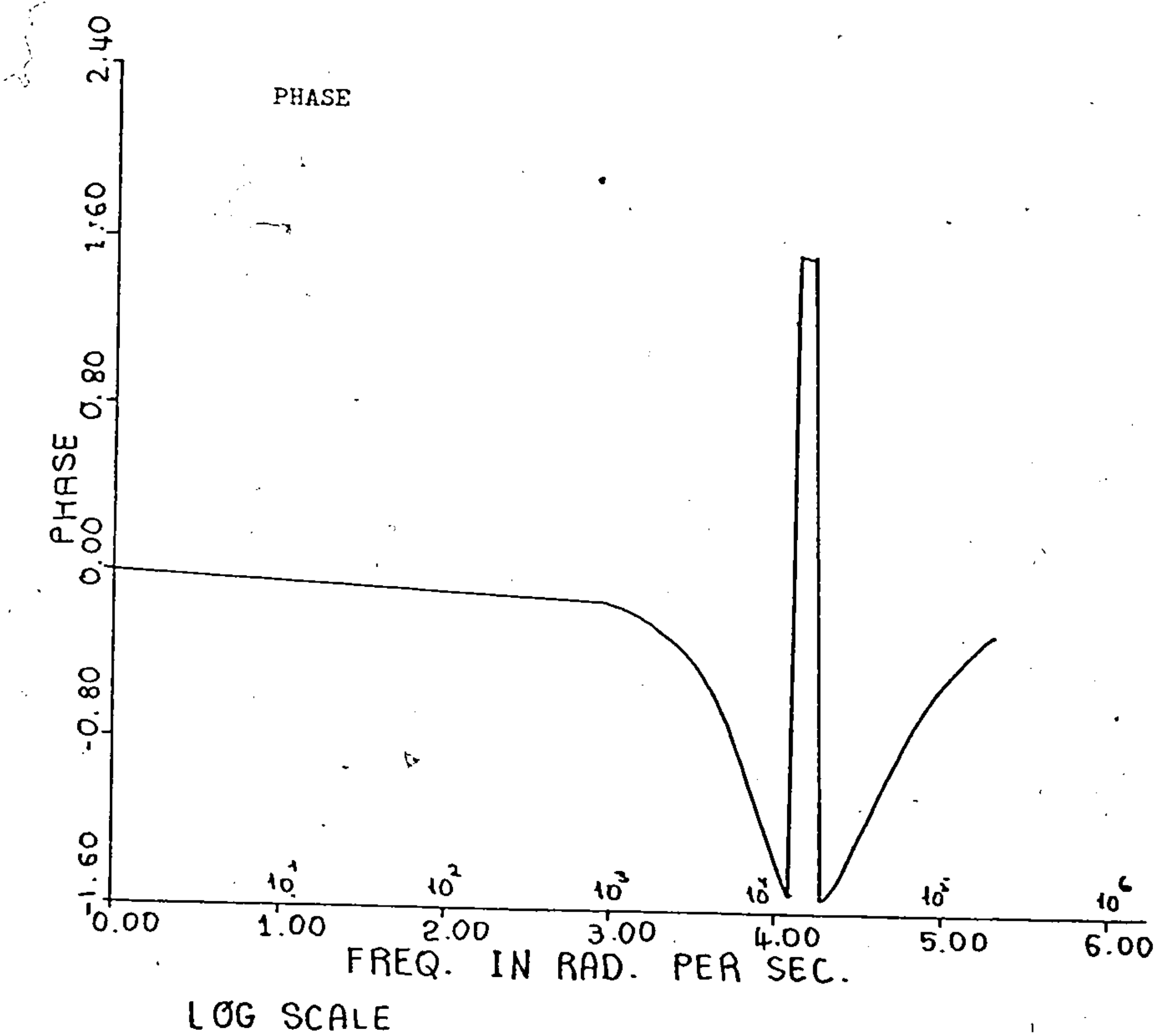
$a_{j1}=0.492725\times10^{5}=a_{j2}$

$b_{j1}=0.402290\times10^{4}=b_{j2}$

Example 2 : Third order low-pass filter

Q = 1.07466

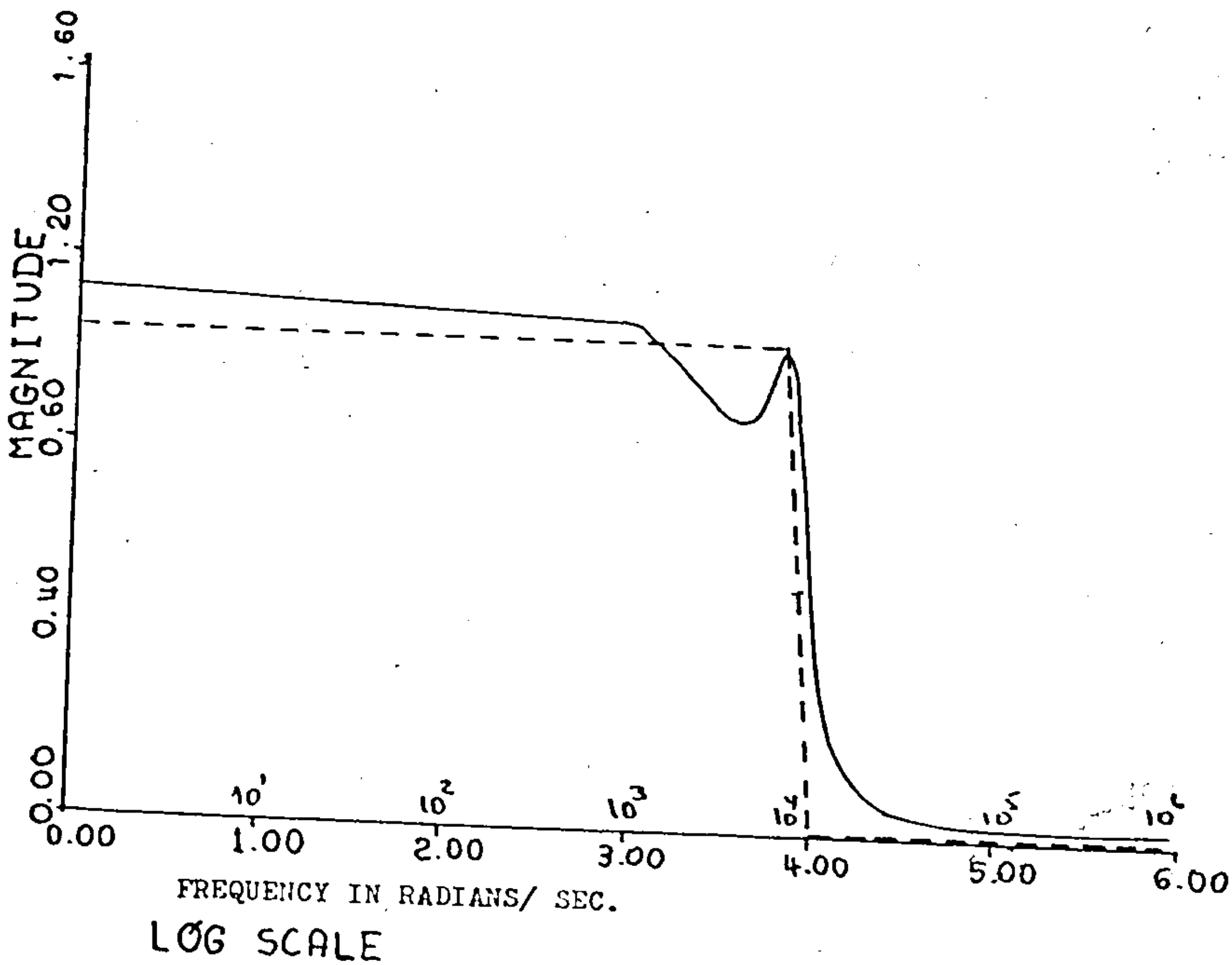


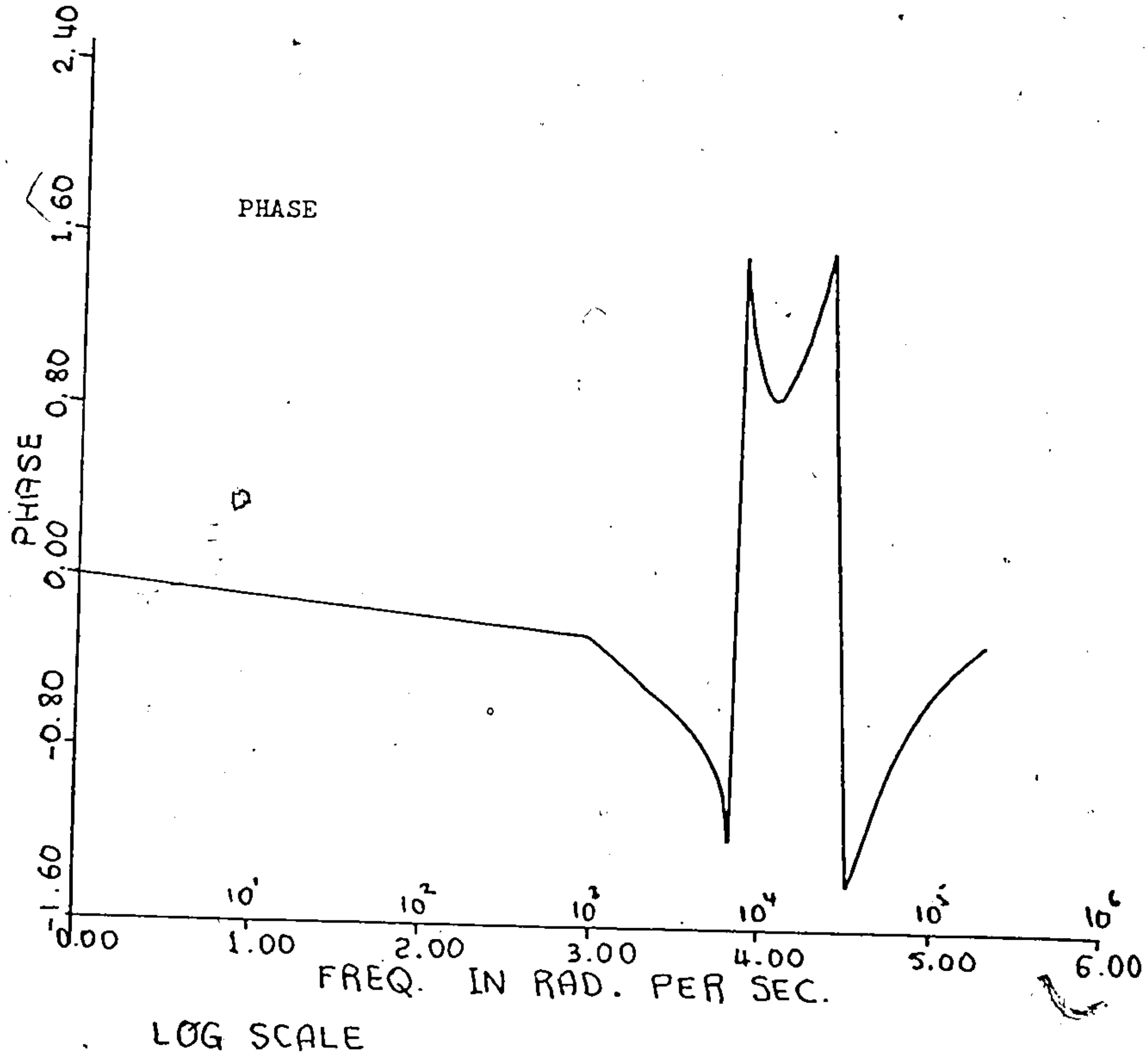

PHASE
PHASE
2.40
1.60
0.80
0.00
-0.80
-1.60
10^1
10^2
10^3
10^4
10^5
10^6
0.00
1.00
2.00
3.00
4.00
5.00
6.00
FREQ. IN RAD. PER SEC.
LOG SCALE

Example 2 : Third order low-pass filter

Q= 0.49738

(30 iterations)

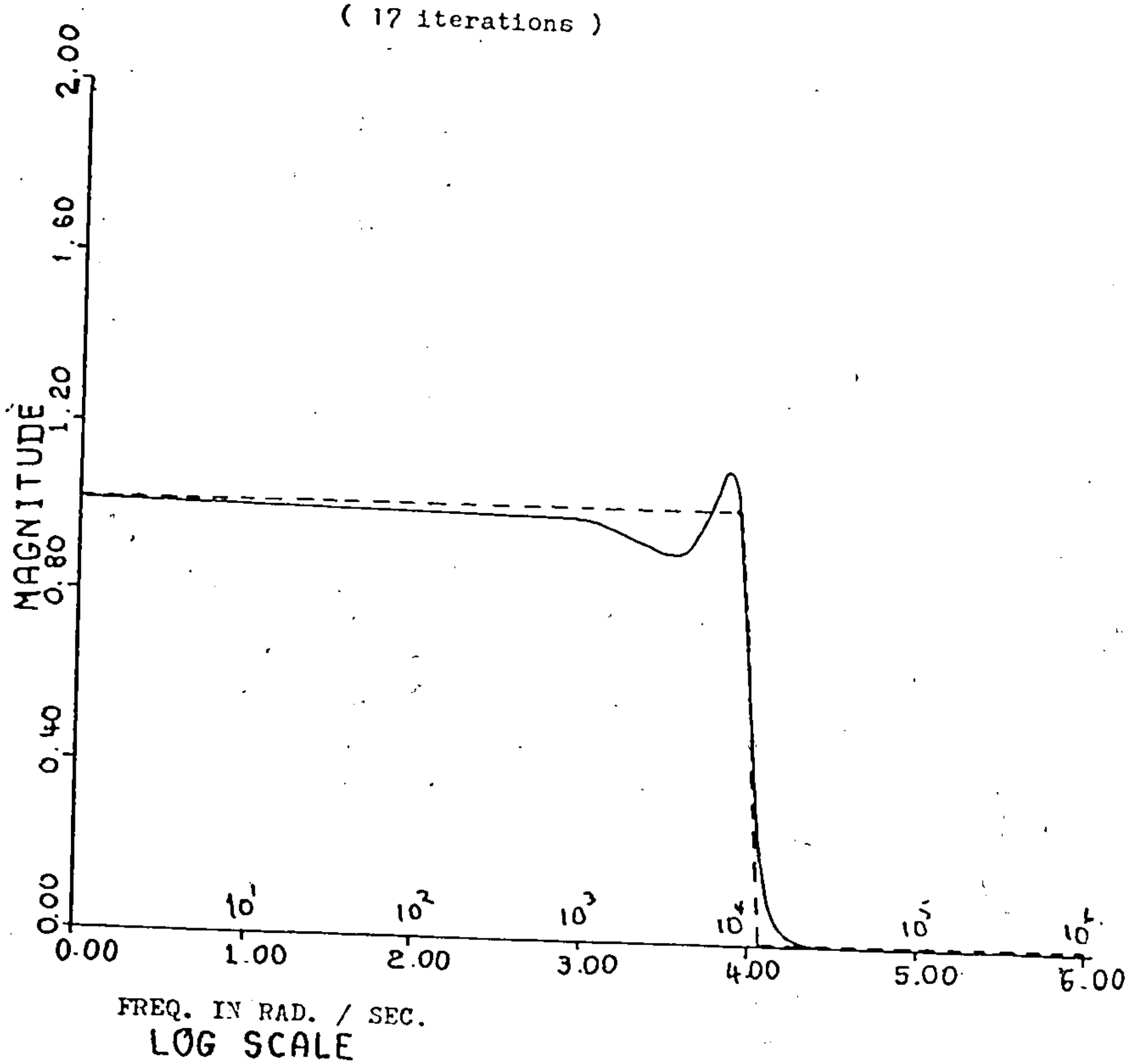


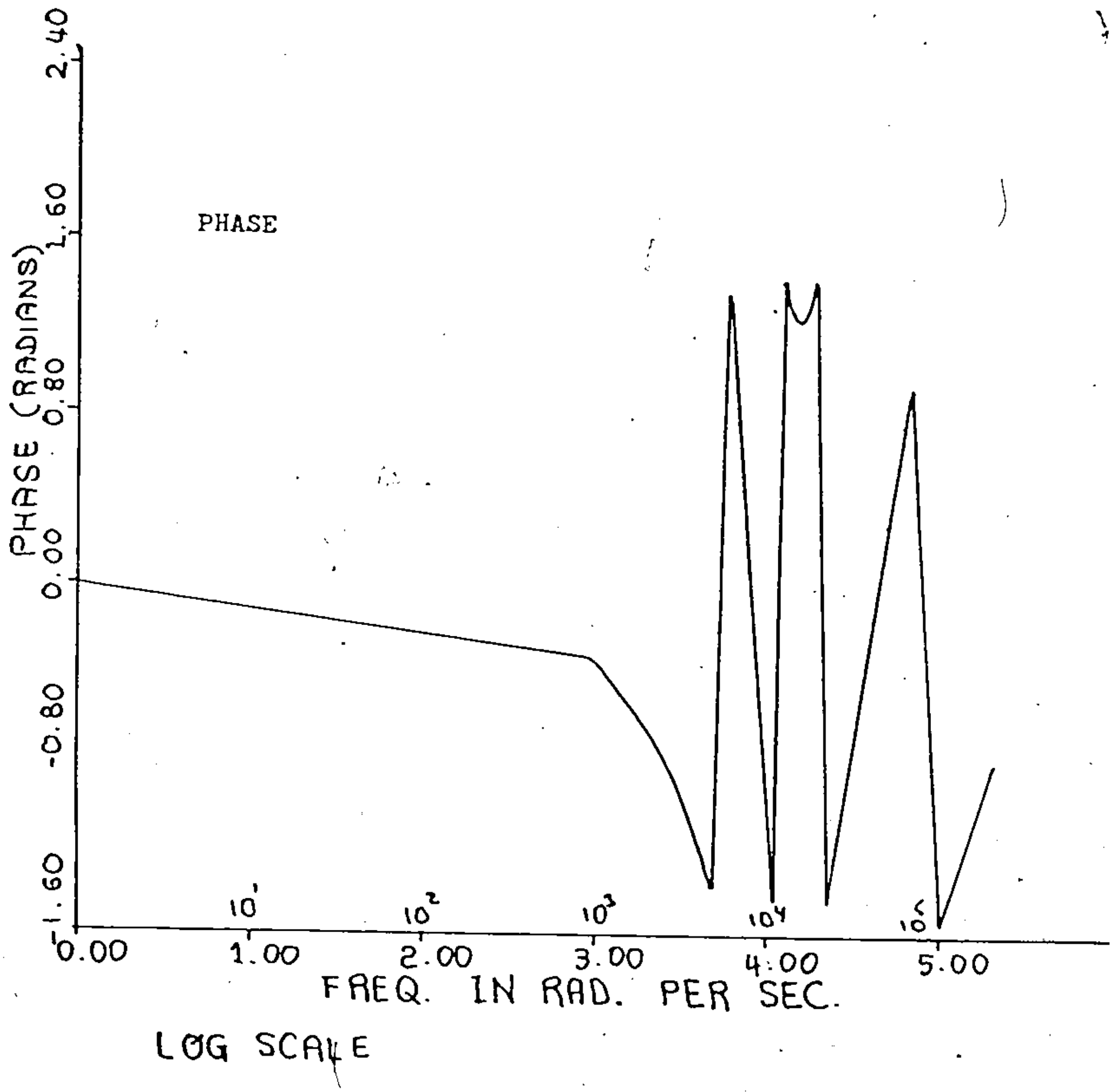

PHASE
PHASE
2.40
1.60
0.80
0.00
-0.80
-1.60
10^1
10^2
10^3
10^4
10^5
10^6
0.00
1.00
2.00
3.00
4.00
5.00
6.00
FREQ. IN RAD. PER SEC.
LOG SCALE

Example 2 : Eighth order low-pass filter

Q = 0.22632

(17 iterations)




PHASE
PHASE (RADIANS)
2.40
1.60
0.80
0.00
-0.80
-1.60
10'
10²
10³
10⁴
10⁵
0.00
1.00
2.00
3.00
4.00
5.00
FREQ. IN RAD. PER SEC.
LOG SCALE

Example 3 : Design a high-pass filter having the following specifications

$w_a=0.0,9000(1000);$	$Y^d=0.0$
$w_a=10^4$;	$Y^d=0.707$
$w_a=13000,30000(3000);$	$Y^d=1.0$

1- λ =scale factor = 1.0/30000

2- $w_i=\frac{2}{\pi}\tan^{-1}(w_{ai}\cdot\lambda)$.

3- Frequencies between 0.5 and 1.0 times the Nyquist rate, which correspond to 30000 rad/sec., and infinite rad./sec. are selected as follows

$w_i=0.505,1.0(0.05);$	$Y^d=1.0$

4-Proceeding as in example 2 , Ψ_0 is chosen to be

$$\Psi_0=(0.4,0.4,0.4,0.4;0.4,0.4)^T$$

For K=L=1, and after 60 iterations , the results for the analog filter desired are

$Q=1.0060$

$A=0.8941$

$a_1=0.6795\times10^5$

$b_1=0.1166\times10^{10}$

$c_1=0.4069\times10^5$

$d_1=0.8354\times10^9$

$a_{j1}=0.49831\times10^{2}$

$b_{j1}=0.29753\times10^{5}$

Poles $(-0.20348\times10^{5}\pm j0.34195\times10^{4})$
$(-b_{j1})$

Zeros $(-0.33979\times10^{5}\pm j0.20529\times10^{5})$
$(-a_{j1})$.

For the same filter specifications, except that at

$w_a=10^4$; $Y^d=0.50$

and K=3 ,L=2 is chosen.

ψ_0 was chosen as the final value obtained from the previous case

ψ_0 =(-0.31074,0.56274,0.73113,0.01135, (repeated 3 times , since K=3); 1.50023, 0.14744, (repeated 2 times,since L=2)).

After 30 iterations the results obtained are.

$Q=0.43007$

$A=0.92021$

$a_1=0.623451\times10^{5}$

$b_1=0.99282\times10^{9}$

$c_1=0.15352\times10^5$

$d_1=0.25867\times10^9$

$a_2=0.61965\times10^5$

$b_2=0.97905\times10^9$

$c_2=0.65973\times10^5$

$d_2=0.11172\times10^{10}$

$a_3=0.62762\times10^5$

$b_3=0.10077\times10^{10}$

$c_3=0.72765\times10^5$

$d_3=0.13466\times10^{10}$

$a_{j1}=0.3982\times10^3$

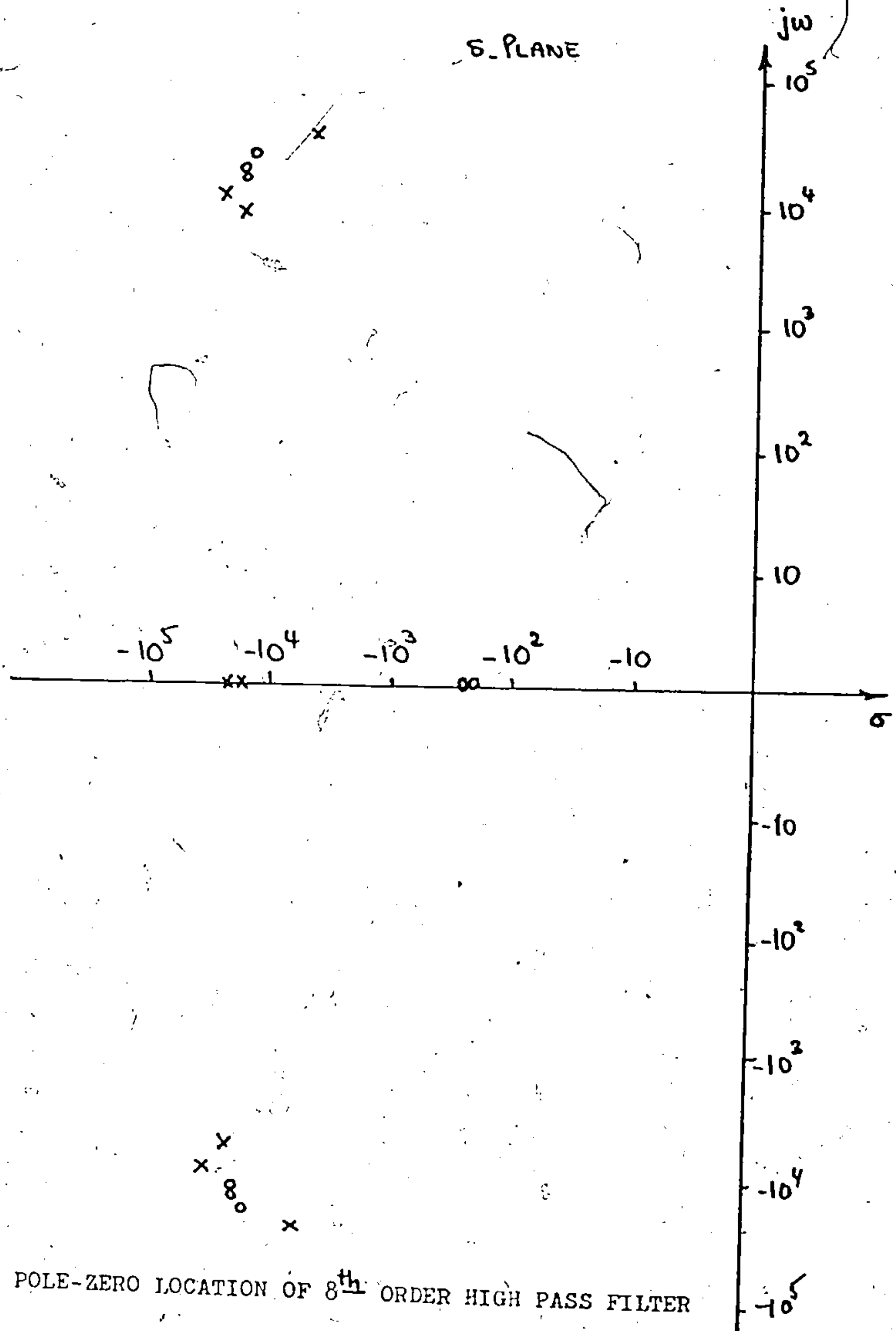
$b_{j1}=0.2494\times10^5$

$a_{j2}=0.3982\times10^3$

$b_{j2}=0.2366\times10^5$

Poles
$(-0.76758\times10^4 \pm j0.45925\times10^4)$
$(-0.32987\times10^5 \pm j0.43749\times10^4)$
$(-0.36382\times10^5 \pm j0.47948\times10^4)$
$(-b_{j1}$, and $-b_{j2})$

Zeros
$(-0.311725\times10^5 \pm j0.1413\times10^5)$
$(-0.309824\times10^5 \pm j0.5593\times10^4)$
$(-0.313809\times10^5 \pm j0.4789\times10^4)$
$(-a_{j1}$, and $-a_{j2})$

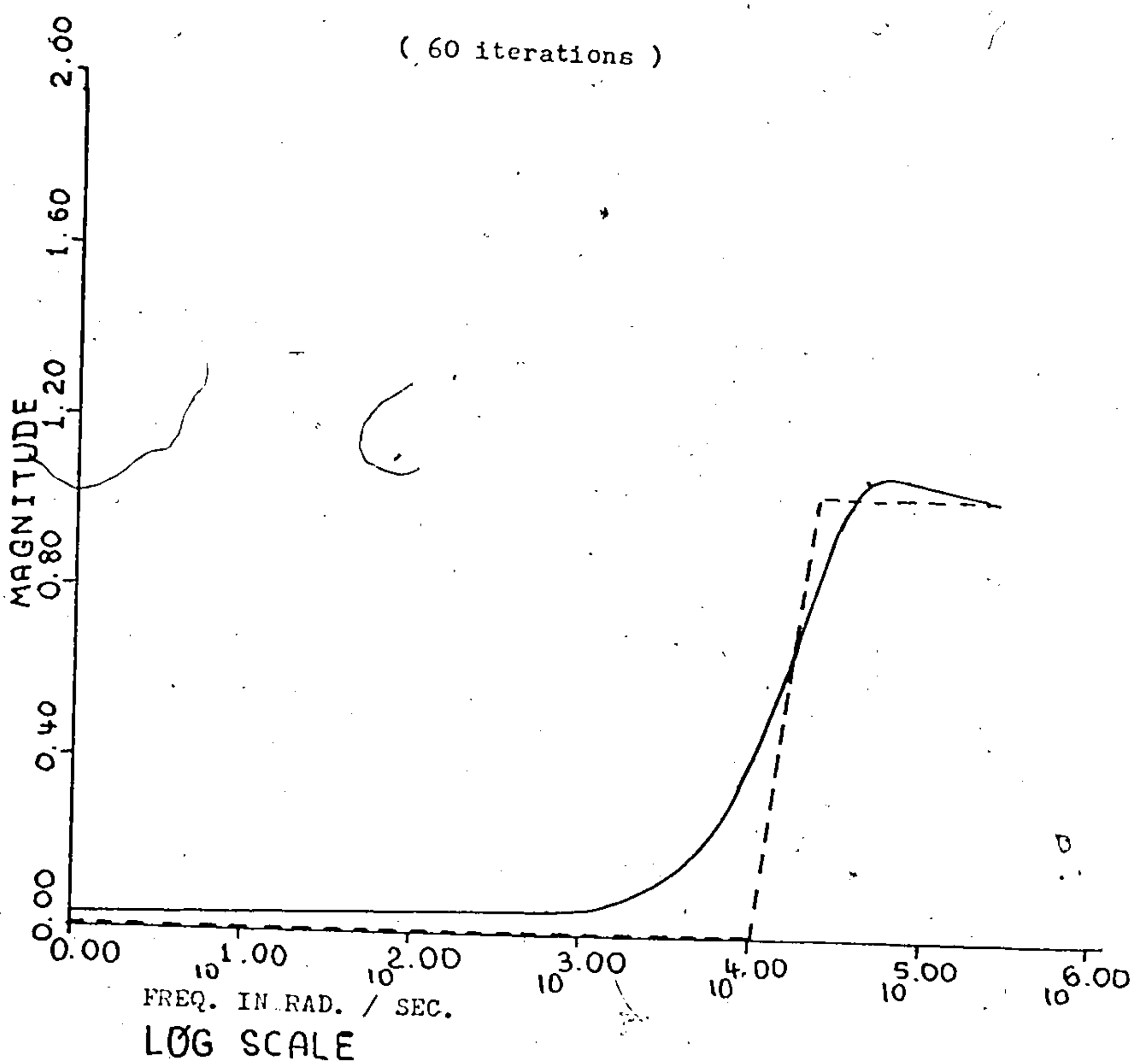


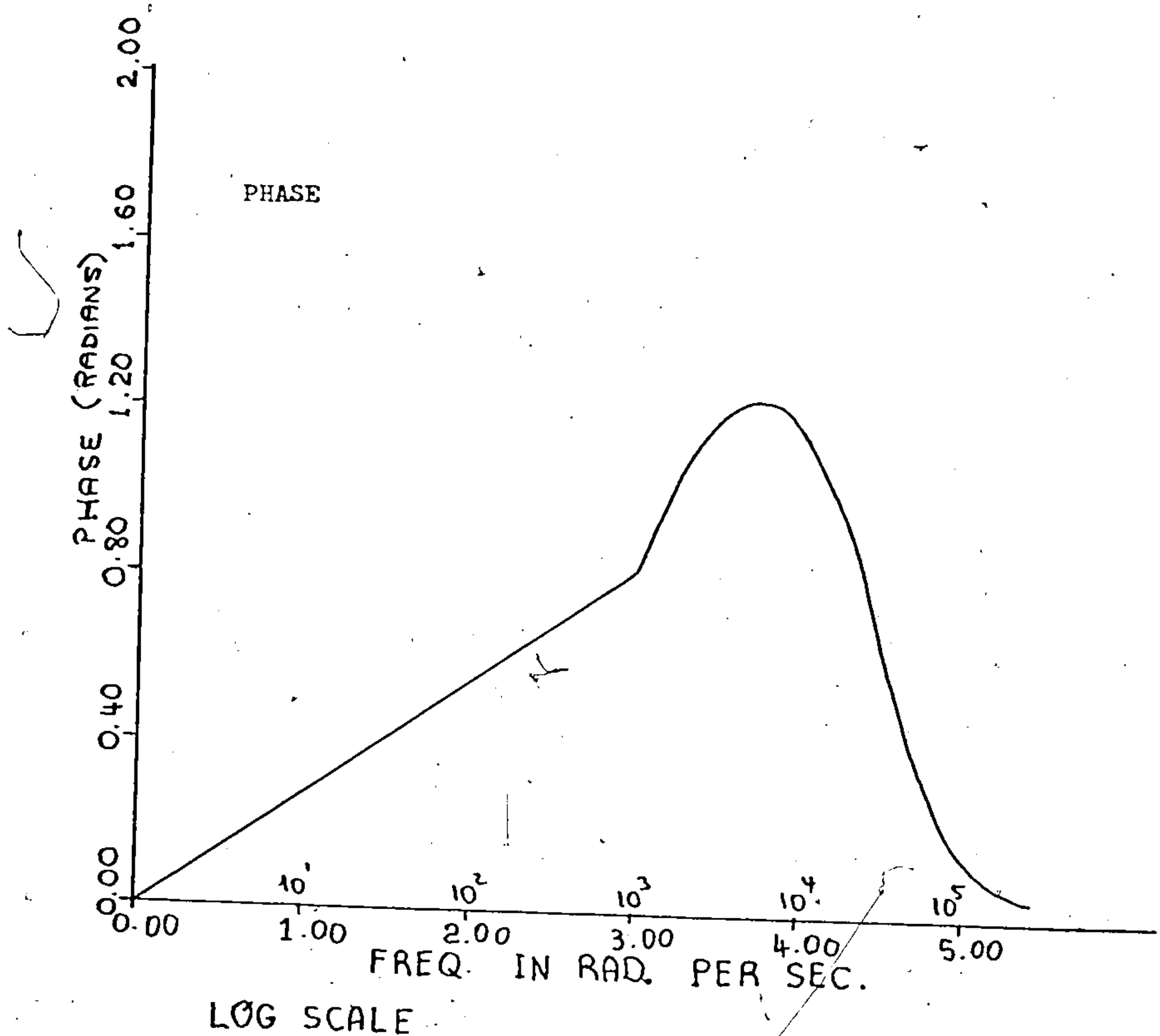
POLE-ZERO LOCATION OF $8^{\underline{th}}$ ORDER HIGH PASS FILTER

Example 3 : Third order high-pass filter

Q=1.0060

(60 iterations)

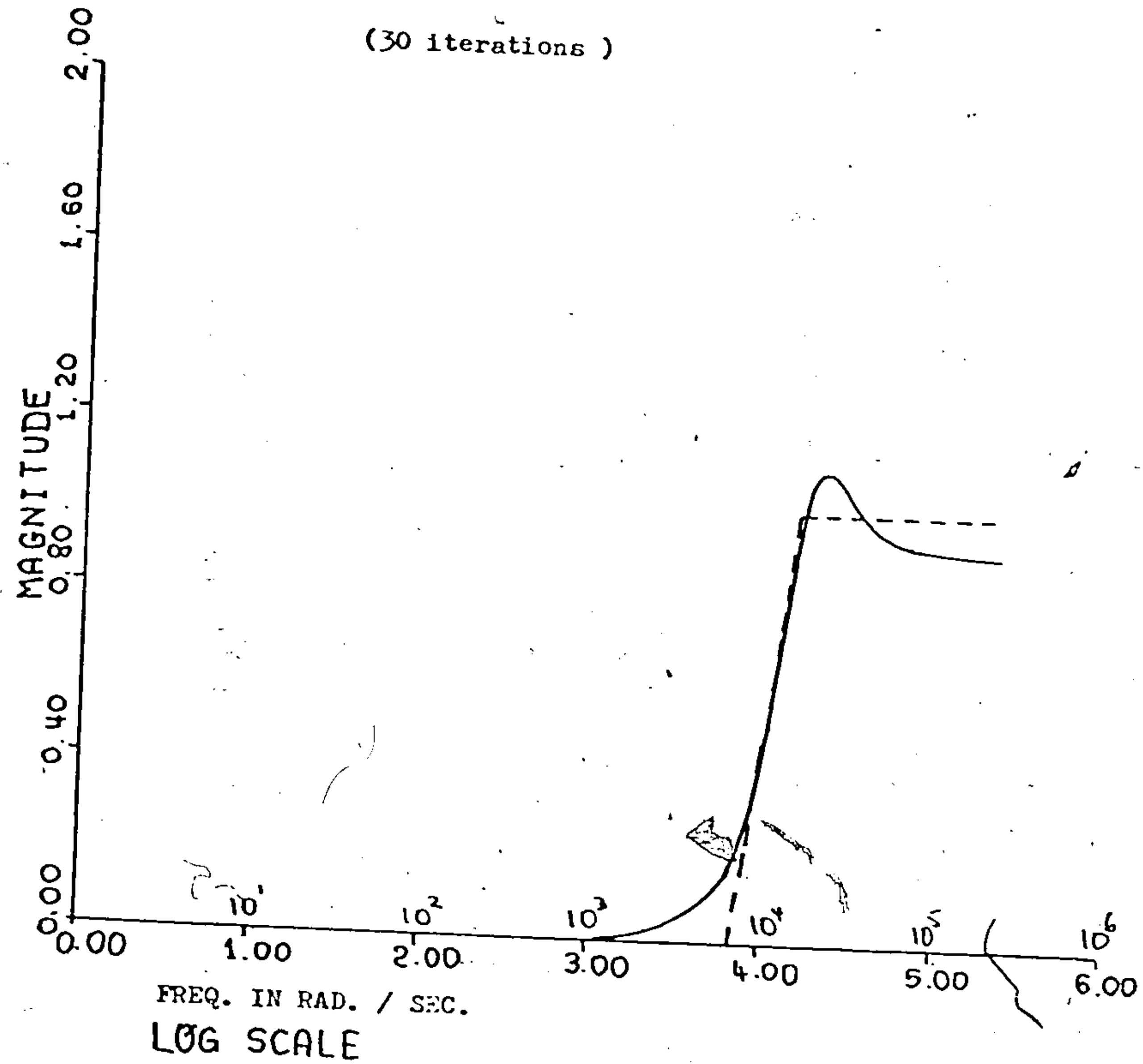


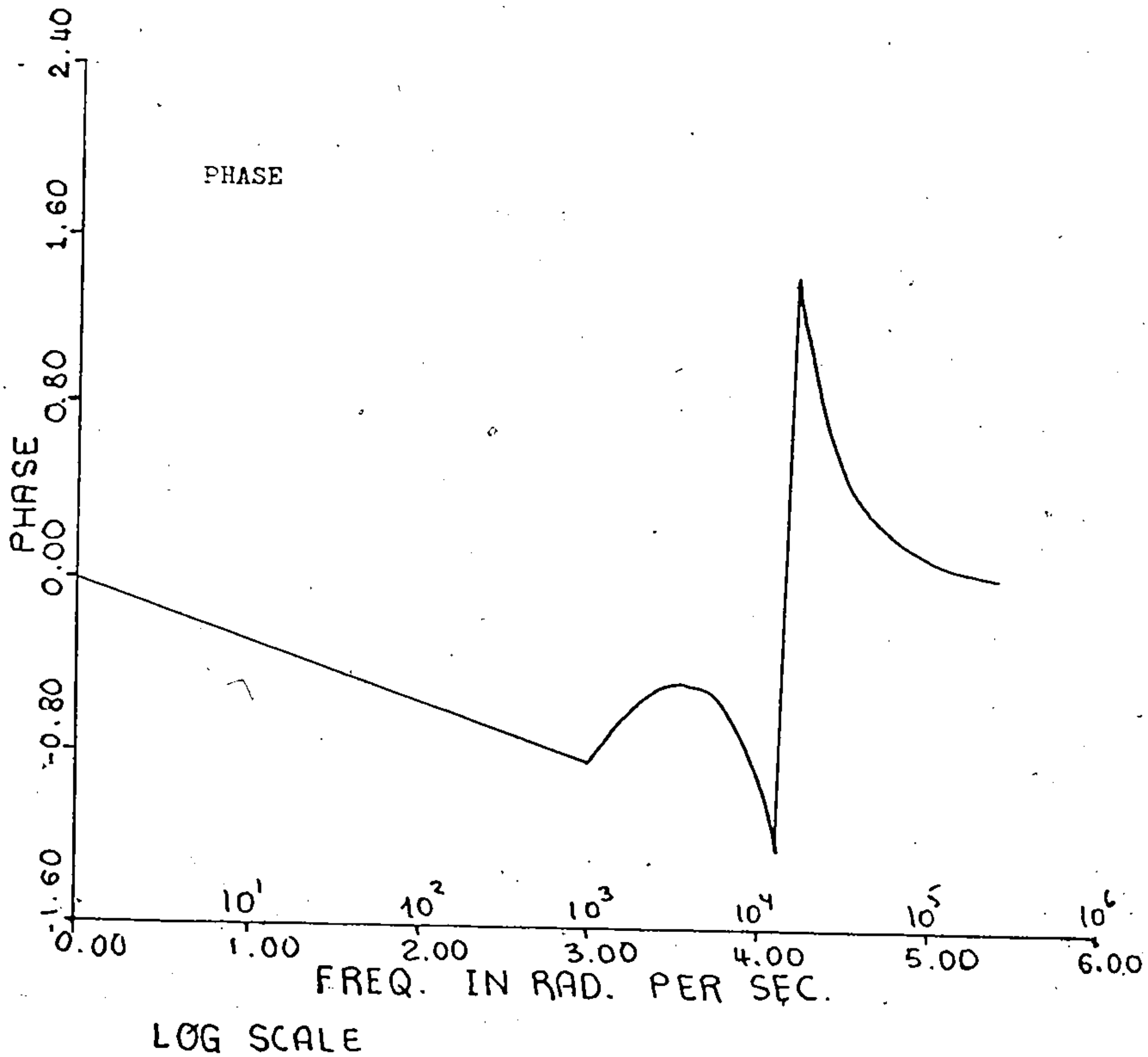

PHASE
PHASE (RADIANS)
2.00
1.60
1.20
0.80
0.40
0.00
0.00
1.00
2.00
3.00
4.00
5.00
10'
10²
10³
10⁴
10⁵
FREQ. IN RAD. PER SEC.
LOG SCALE

Example 3 : Eighth order high-pass filter

Q = 0.43007

(30 iterations)




PHASE
PHASE
2.40
1.60
0.80
0.00
-0.80
-1.60
10^1
10^2
10^3
10^4
10^5
10^6
0.00
1.00
2.00
3.00
4.00
5.00
6.00
FREQ. IN RAD. PER SEC.
LOG SCALE

CHAPTER VI

DESIGN OF DIGITAL FILTERS BY IMPLEMENTING CONSTRAINTS ON AN ANALOG FILTER

6.1 THE METHOD

Consider an analog filter having the transfer function

$$Y(s) = A \prod_{k=1}^{K} \frac{s^2 + a_k^2 . s + b_k^2}{s^2 + c_k^2 . s + d_k^2} \qquad (6.1)$$

The coefficients are squared to ensure that they always remain positive , and hence the poles and zeros are confined to the left-half of the s-plane.

If the left-half of the s-plane is mapped onto the unit circle in the z-plane , and the poles and zeros of the transfer-function given by eq.(6.1) are mapped into the z-plane, then the mapped function will have its poles and zeros confined inside the unit circle.

The bilinear transformation which maps the jw-axis ,and the left-half of the s-plane onto a unit circle in the z-plane is given by

$$s = \frac{z - 1}{z + 1} \qquad (6.2)$$

Hence , the transfer function of the recursive digital filter that has its poles and zeros confined inside the unit circle is obtained by substituting eq.(6.2) in eq.(6.1) .

Therefore ;

$$Y(z,X) = A \prod_{k=1}^{K} \frac{z^2(1+a_k^2+b_k^2) + z(-2+2b_k^2) + (1-a_k^2+b_k^2)}{z^2(1+c_k^2+d_k^2) + z(-2+2d_k^2) + (1-c_k^2+d_k^2)} \qquad (6.3)$$

$$= A.H(z,\psi)$$

where, $X=(a_k,b_k,c_k,d_k;A)^T$

and $\psi=(a_k,b_k,c_k,d_k)^T$

6.2 Summary of the subroutine FUNCT(N,X,Q,DQ)

1- Calculate H_i $\qquad i=1,2,\ldots\ldots,M$

2- Calculate

$$A^* = \frac{\sum_{i=1}^{M} |H_i| \; Y_i^d}{\sum_{i=1}^{M} |H_i|^2}$$

3- Calculate $E_i = A^* |H_i| - Y_i^d$ $\qquad i=1,2,\ldots,M$

4- Calculate $Q = \sum_{i=1}^{M} E_i^2$

5- Calculate $\dfrac{\partial |H_i|}{\partial \psi_n}$ $\quad n=4K$

$$\frac{\partial |H_i|}{\partial a_k} = |H_i| . Re\left(\frac{2a_k z^2 - 2a_k}{N} \right)$$

$$\frac{\partial |H_i|}{\partial b_k} = |H_i| . Re\left(\frac{2b_k z^2 + 4b_k z + 2b_k}{N} \right)$$

$$\frac{\partial |H_i|}{\partial c_k} = |H_i| . Re\left(\frac{2c_k z^2 - 2c_k}{D} \right)$$

$$\frac{\partial |H_i|}{\partial d_k} = |H_i| . Re\left(\frac{2d_k z^2 + 4d_k z + 2d_k}{D} \right)$$

where, $N = z^2(1 + a_k^2 + b_k^2) + z(-2 + 2b_k^2) + (1 - a_k^2 + b_k^2)$

$D = z^2(1 + c_k^2 + d_k^2) + z(-2 + 2d_k^2) + (1 - c_k^2 + d_k^2)$

6.3 EXAMPLE

Consider the specifications of the ideal low-pass filter given in sec.3.9.

In this case a second order filter is considered , and the parameter vector is initially chosen as

$$\psi_o = (0.1, 0.1, 0.1, 0.1)^T$$

After 100 iterations

$$Q = 0.53475$$

$$A^* = 0.1368$$

Poles $(0.8957 \pm j0.19554)$

Zeros $(0.84124 \pm j0.54045)$

If K=2 is now chosen , and the final value of ψ is taken as the initial value in this case. After 100 iterations

$$Q = 0.032563$$

$$A^* = 0.031198$$

Poles $(0.86281 \pm j0.13583)$

$(0.93122 \pm j0.27786)$

Zeros $(0.92596 \pm j0.37744)$

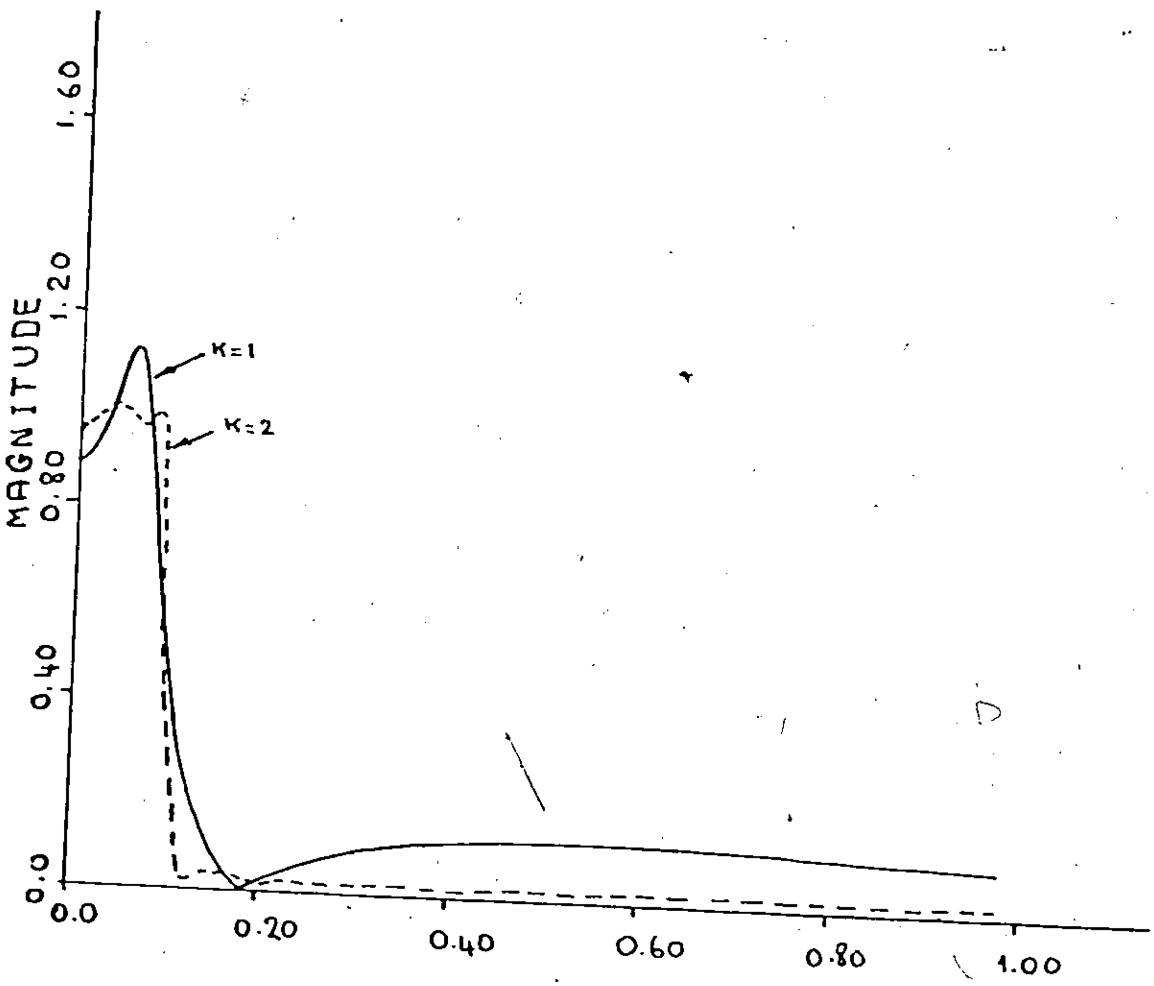
$(0.67643 \pm j0.73636)$

6.4 DISCUSSION

The method discussed in this chapter has two advantages over the method previously discussed in chapter III:

1-It requires less computer time per iteration.

2-The poles are not constrained to be complex or real, but have the freedom to assume whatever configuration is best for the desired specifications.



FREQUENCY IN FRACTIONS OF THE NYQUIST FREQUENCY

CHAPTER VII

AN OUTLINE OF A THEORY FOR THE DESIGN OF A DIGITAL FILTER THAT WILL MEET A REQUIRED MAGNITUDE AND PHASE RESPONSE

7.1 INTRODUCTION

The method described here is that of the use of unconstrained minimization for the design of a recursive digital filter that will meet a required magnitude and phase response.

7.2 FILTER FORM

The filter form used is that of a cascade for reasons previously mentioned.

$$Y(z_i) = \prod_{k=1}^{K} \frac{a_k z_i^2 + b_k z_i + c_k}{(z - p_k e^{j\theta_k})(z - p_k e^{j\theta_k})} \prod_{j=1}^{L} \frac{z - a_j}{z - b_j} \qquad (7.1)$$

7.3 PHASE RESPONSE

Let ϕ_i be the phase shift introduced by the filter at the frequency w_i .

$$\phi_i = \mathrm{TAN}^{-1}(\mathrm{Im}.Y(z_i)/\mathrm{Re}.Y(z_i)) \qquad (7.2)$$

7.4 SQUARE-ERROR CRITERION

Let Y_i^d be the desired magnitude at w_i , and ϕ_i^d be the desired phase at w_i. w_i, is expressed in fractions of the Nyquist frquency.

Choose an error criterion of the form

$$Q(z_i,\psi) = \sum_{i=1}^{M} \lambda_1 (|Y(z_i,\psi)| - Y_i^d)^2 + \sum_{i=1}^{M} \lambda_2 (\phi_i(z_i,\psi) - \phi_i^d)^2 \quad (7.3)$$

where ψ is the vector of unknown parameters, λ_1 and λ_2 are constants ≥ 0.

7.5 CONSTRAINTS

For stability reasons the poles of the digital filter should lie inside the unit circle of the z-plane. To take care of that we choose a new set of variables related to the variables p_k and b_j as follows

$$\begin{aligned} p_k &= \sin^2 \beta_k \\ b_j &= \sin \theta_{pj} \end{aligned} \quad (7.4)$$

Therefore,

$$Y(z_i) = \prod_{k=1}^{K} \frac{a_k z_i^2 + b_k z_i + c_k}{z_i^2 - 2\sin^2\beta_k . \cos\theta_k . z_i + \sin^4\beta_k} \prod_{j=1}^{L} \frac{z_i - a_j}{z_i - \sin\theta_{pj}} \quad (7.5)$$

Another method of implementing constraints on the poles is that given in sec.6.1.

If we assume that we have an analog filter of the form

$$Y(s_i) = A \prod_{k=1}^{K} \frac{s^2 + a_k s + b_k}{s^2 + c_k^2 s + d_k^2} \quad (7.6)$$

by substituting $s = \dfrac{z-1}{z+1}$ in eq.(7.6) we get the required digital filter with poles constrained inside the unit circle.

$$Y(z,\psi_n) = A \prod_{k=1}^{K} \frac{z^2(1+a_k+b_k) + z(-2+2b_k) + (1-a_k+b_k)}{z^2(1+c_k^2+d_k^2) + z(-2+2d_k^2) + (1-c_k^2+d_k^2)} \qquad (7.7)$$

7.6 TO OBTAIN $\dfrac{\partial Q}{\partial \psi_n}$

$$\frac{\partial Q}{\partial \psi_n} = 2\lambda_1 \sum_{i=1}^{M} (|Y(z_i,\psi_n)| - Y_i^d) \frac{\partial |Y(z_i,\psi_n)|}{\partial \psi_n} + 2\lambda_2 \sum_{i=1}^{M} (\phi(z_i,\psi_n) - \phi_i^d) \frac{\partial \phi(z_i,\psi_n)}{\partial \psi_n} \qquad (7.8)$$

where n= the number of parameters .

Since , $\phi_i = \tan^{-1}(\mathrm{Im}(Y(z_i,\psi_n))/\mathrm{Re}(Y(z_i,\psi_n)))$

Therefore,

$$\frac{\partial \phi_i}{\partial \psi_n} = \frac{1}{|Y_i|^2} \left(\mathrm{Re}(Y_i).\mathrm{Im}\left(\frac{\partial Y_i}{\partial \psi_n}\right) - \mathrm{Im}(Y_i).\mathrm{Re}\left(\frac{\partial Y_i}{\partial \psi_n}\right) \right) \qquad (7.9)$$

---

* <u>Note</u> λ_1 and λ_2 are choosen such that the phase and magnitude criterions are of the same order during the minimization procedure. This is usually done in an adaptive manner ,(see example at the end of this chapter).

7.7 SUMMARY OF THE SUBROUTINE FUNCT(N,X,Q,DQ)

1- Calculate Y_i $\quad i=1,2,\ldots\ldots,M$

2- Calculate ϕ_i $\quad i=1,2,\ldots\ldots,M$

3- Calculate

$$E1_i = |Y_i| - Y_i^d$$

$$E2_i = \phi_i - \phi_i^d \qquad i=1,2,\ldots,M$$

4- Calculate

$$Q = \lambda_1 \sum_{i=1}^{M} E1_i^2 + \lambda_2 \sum_{i=1}^{M} E2_i^2$$

5- Calculate $\dfrac{\partial Y_i}{\partial \psi_n}$

n=number of parameteters.

6- Calculate

$$\frac{\partial Q}{\partial \psi_n} = 2\lambda_1 \sum_{i=1}^{M} |Y_i| \,.\mathrm{Re}\left(\frac{\partial Y_i}{\partial \psi_n}\right).E1_i + 2\lambda_2 \sum_{i=1}^{M} \frac{E2_i}{|Y_i|^2} \left(\mathrm{Re}(Y_i)\frac{\partial \mathrm{Im}(Y_i)}{\partial \psi_n} - \mathrm{Im}(Y_i)\frac{\partial \mathrm{Re}(Y_i)}{\partial \psi_n}\right)$$

7.8 USE OF A PENALTY FUNCTION[9]

Another approach to this type of problem is as follows :

minimize
$$Q(z_i,X) = \sum_{i=1}^{M} (|Y(z_i,X)| - Y_i^d)^2 \qquad (7.10)$$

subject to the set of constraints

$$g_i(X) = \sum_{i=1}^{M} \epsilon - (\phi_i^d - \phi_i) \geq 0 \qquad (7.11)$$

where ϵ is the desired error in the phase.

The constraints given by eq.(7.11) could be enforced by adding a penalty function to eq.(7.10) such that if any of the constraints given by eq.(7.11)are violated,the penalty function would assume a large value,otherwise it assumes a small , or zero value .

The simplest choice of the penalty function is the function[9]

$$\sum_{i=1}^{M} |g_i(X)| - g_i(X)$$

For this choice the modified error function is

$$Q(z_i,X) = \sum_{i=1}^{M} (|Y(z_i,X)| - Y_i^d)^2 + r \sum_{i=1}^{M} |g(z_i,X)| - g(z_i,X) \qquad (7.12)$$

where $r > 0$, is a constant. The choice of r is critical: if it is too small the constraints will not be enforced , if too large , inaccuracies will prevent convergence.

7.9 EXAMPLE

Design a differentiator having the following specifications

$$Y_i^d = w_i$$

$$\phi_i^d = w_i \times \pi$$

This problem is solved using the criterion function in sec.7.4, and the filter form is taken as that of eq.7.5 .

A third order filter is considered.

ψ_0 is taken as

$$\psi_0 = (0.4, 0.4, 0.4, 0.4; 0.4, 0.4)^T$$

and $\lambda_1 = 1$, $\lambda_2 = 0.0$.

After 30 iterations , λ_2 is changed to 0.5 , and 100 iterations are carried out.

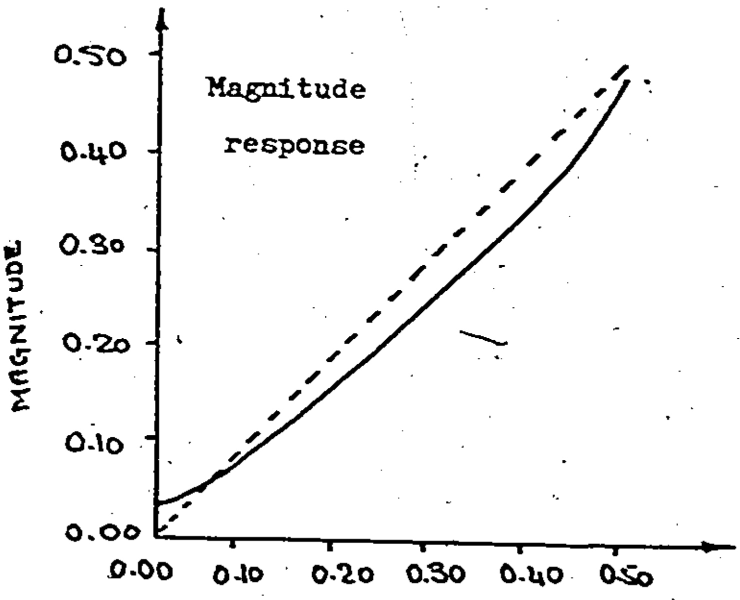
λ_2 is again changed to 0.2, and after 150 further iterations,

$$Q = 6.7776$$

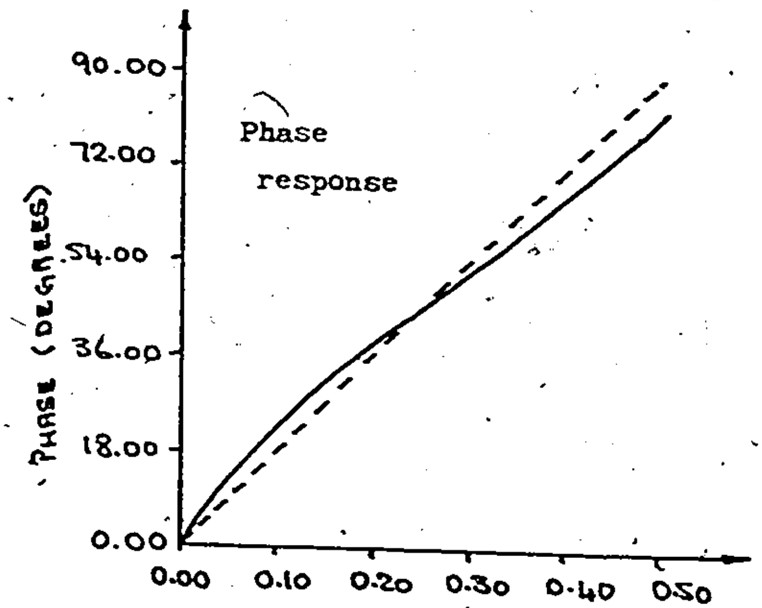
$$a_1 = -0.6495$$

Poles $(0.10425 \times 10^{-3} \pm j0.31718 \times 10^{-4})$
(-0.78785)

Zeros $(0.4915 \times 10^{-1} \pm j0.49671)$
(0.67119)



Freq. in fractions of the Nyquist freq.



Freq. in fractions of the Nyquist freq.

(Dotted curve : desired response ;

Solid curve : designed response)

CONCLUSION

In general design of digital or analog filters , there are two basic techniques available. The first is to derive an analytic expression for the required pole/zero configuration, the second is to use an iterative procedure to approach the desired configuration . It is this second approach that has been investigated in this thesis.

The work utilizes the minimization technique of Fletcher and Powell, and introduces three methods of implementing constraints on the pole-zero location of a recursive digital filter for stability and minimum phase constraint.

The three methods are :

(1) <u>The method introduced by Steiglitz</u> :

This method has the disadvantage that a search for the pole-zero location , to meet required magnitude specifications, is carried in the infinite z-plane, and hence for a bad choice of the starting point convergence might be very slow.

(2) <u>The method of variable transformation</u> :

This improves on the previous method, in the sense that the search is carried out in the unit circle where the poles and zeros are required to be located.

This method has been tested on three types of filters; and proves to give good results in a much fewer number of iterations than obtained with method (1).

Problems were solved using only a specified number of iterations, since it has been found that increasing the iterations to convergense does not significantly reduce the value of Q.

(3) The method of complex variable mapping :

This method makes use of the fact that it is easier to implement constraints on the pole-zero location in the s-plane than it is in the z-plane.

This method , though it does not converge as fast as the second method requires less computer time per iteration.

The work has been extended to include the design of continuous filters, also utilizing the Fletcher-Powell minimization technique.

Two methods were introduced for analog filter design :

(1) The variable transformation method :

This method is concerned with required magnitude specifications specified over a finite range of frequencies.

(2) The bilinear transform :

This method is a more powerful method for designing analog filters having required specifications over the infinite frequency axis.

At the end of this work a method has been introduced whereby it is possible to design filters having required magnitude and phase specifications.

Further work along these lines might take into account :

(1) Arbitrary weighting of the errors at different specification points.

(2) A criterion for the choice of the multipliers λ_1 and λ_2 defined in chapter VII.

(3) It has been found that it takes fewer number of iterations to design a filter where magnitude specifications do not change abrubtly. Hence, a useful investigation would be a search for a suitable transformation to transform sharp cut-off filters to a domain where the magnitude response is more smooth.

REFERENCES

1- K.Steiglitz , Computer-aided design of recursive digital filters , IEEE Transaction on Audio and Electroacoustics , vol.Au-18, no.2,June 1970.

2- B.Gold and M.Radar , Digital Processing of Signals ,McGraw-Hill ,1969.

3- D.P.Lindorff , Theory of sampled-data control systems , John Wiley and Sons,Inc., 1965.

4- L.S.Lasdon, Optimization theory for large systems. Macmillan Series , 1970.

5- R.Fletcher and M.J.D.Powell,A rapidly convergent descent method for minimization , Computer J. , vol.6, no.2, pp.163-168, 1963.

6- System/360 Scientific subroutine package (360-CM-03X), version III, Programmer's manual, IBM Data Processing Division, White Plains, N.Y., Document H20-0205-3,1968.

7-M.J.Box, A comparison of several current optimization methods, and the use of transformations in constrained problems, Computer J. vol.9, pp.67-77; May, 1966.

8- C.R.Wylie, Jr., Advanced Engineering Mathematics, McGraw-Hill, Third Edition, 1966.

9- F.Kuo and W.G.Magnuson, Computer oriented circuit design, Prentice-Hall, 1969.

APPENDIX

Programs and subroutines used

```
      ENTRY
C     .... [illegible]
C.....DESIGN [illegible]
      DIMENSION YS(30),YPS(30),PHASES(30),YS(30),POLER(4),POLEI(4),
     1 ZEROR(4),ZEROI( )
      DOUBLE PRECISION X(16),[illegible](16),YP(30),[illegible](30),WC(124),[illegible],A(3),[illegible](3),
     1 C(3),B(3),AJ(2),BJ(2),DSIN,PY,Y(30),DCOS,DABS,AR,ARG,DREAL,DI[illegible]G,
     1 DATA[illegible],PHASE(30)
      COMPLEX*16 Z(30),C[illegible]C PLX,CDEXP,H(30),R1,R2,POLE1(3),POLE2(3)
     1 ,ZE[illegible]1( ),ZER2( ),[illegible],DCE JG,CDSQRT
      COMMON /B1/Z,A
      COMMON /B2/YP,[illegible]
      COMMON /B3/ ,[illegible]L
      COMMON /B4/[illegible]T
      PY=3.141592653589[illegible]9
      EXTERNAL FUNCT
C
C
C..DIMENSION [illegible]
C   [illegible]
C..[illegible]
C
C.....[illegible] IS SPECIFIED [illegible] THE LIT[illegible] IS THE [illegible]
C ....[illegible] SPECIFICATIONS [illegible]
C
C.................................................................................
C [illegible]
C.....[illegible]
C...LIMIT IS [illegible]
      [illegible]=1
    1 [illegible]=2
      L=2
C  [illegible]=X(1),[illegible]=X(2),[illegible]=X(3),[illegible]=X(4),[illegible]=X(5),[illegible]=X(3),
C  [illegible]=2,[illegible](1)=[illegible](1),[illegible](2)=[illegible](2),[illegible](1)=[illegible](4)..ETC
      Y[illegible](1)=1.
      [illegible](1)=[illegible].
      [illegible] I=1,[illegible]
      [illegible](I)=1.
    2 [illegible](I+1)=[illegible](I)+[illegible]
      Y[illegible](11)=[illegible]
      [illegible](11)=[illegible]1
      [illegible] I=12,21
      [illegible](I)=[illegible].
   12 [illegible](I)=[illegible](I-1)+[illegible]
      [illegible] I=22,2[illegible]
      [illegible](I)=[illegible].
   13 [illegible](I)=[illegible](I-1)+[illegible].1
      IF([illegible])[illegible] TO [illegible]
      [illegible](1)=PY/6.[illegible]
      [illegible](2)=PY/2.[illegible]
      [illegible](3)=[illegible]
      [illegible](4)=PY/2.[illegible]
  600 LIMIT=[illegible]
C
C.................................................................................
C
      [illegible]=[illegible]+2*L
      DO 4 I=1,[illegible]
      CP=DCMPLX(0.D0,[illegible](I)*3.1415900)
    4 Z(I)=CDEXP(CP)
```

```
      EST=0.1
      EPS=1.0/10.**5
      CALL DEMEP(FUNCT,N,X,O,DO,EST,EPS,LIMIT,IER,MC)
      WRITE(6,700) KOUNT
700   FORMAT(5X,'KOUNT=',I5)
      WRITE(6,6)IER
      FORMAT(5X,'IER=',I5)
      WRITE(6,8)O
8     FORMAT(5X,'O=',D20.12)
      WRITE(6,11) DO
11    FORMAT(5X,'DO=',D20.12)
      WRITE(6,100)(I,X(I),I=1,N)
100   FORMAT(5X,'X(',I3,')=',D20.12)
      IF(N.[illegible].0)GO TO [illegible]
      DO 14 I=1,K
      K1=K+I
      K2=2*K+I
      K3=3*K+I
      A(I)=-2.*A(DSI (X(I))**2)*DCOS(X(K1))
      B(I)=DSI (X(I))**2
     C(I)=-2.*(DSI (X(K2))**2)*DCOS(X(K3))
      D(I)= SI (X(K2))**2
      WRITE(6,27) I,A(I),I,B(I),I,C(I),I,D(I)
27    F    T(5X,' (',I3,')=',D20.12/5X,'B(',I3,')=',D20.12/5X,'C(',I3,
     1 ')=',D20.12/ 5X,'D(',I3,')=',D20.12)
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
15    FORMAT(5X,'ZPL1(',I3,')=',[illegible])
16    FORMAT(5X,'ZPL2(',I3,')=',[illegible])
17    FORMAT(5X,'ZMI1(',I3,')=',[illegible])
1     FORMAT(5X,'ZMI2(',I3,')=',[illegible])
14    CONTINUE
      J=1
      IF(L.[illegible].0)GO TO [illegible]
      LL=[illegible]
      LK=LL+[illegible]
      DO 1  JJ=LL,LK
      [illegible]=JJ
      [illegible]=JJ+L
      [illegible](J)=[illegible]
      [illegible](J)=[illegible]
      [illegible]
2     [illegible]
      [illegible]
      [illegible]
10    J=J+1
      CONTINUE
      DO    I=1,
      XS(I)=SNGL(X(I))
30    YPS(I)=SNGL(Y(I))
      DO 25 I=1,
      Y(I)=[illegible]
      ARG=[illegible]
25    [illegible]
```

```
      WRITE(6,71)
71    FORMAT(4X,'I',20X,'Y(I)',30X,'W(I)',30X,'PHASE(I)')
      WRITE(6,72)(I,Y(I),W(I),PHASE(I),I=1, )
72    FORMAT(4X,I3,6X,D20.12,21X,D20.12,15X,D20.12)
      DO 21 I=1,
      ...(I)=S...L(...(I))
21    YS(I)=SNGL(Y(I))
      WRITE(6,24)
24    FORMAT(1H1,'PLOTTING OF THE GIVEN FILTER RESPONSE')
      CALL PLOTR(WS,YDS, )
      WRITE(6,26)
26    FORMAT(1H1,'PLOTTING OF THE DESIGNED FILTER RESPONSE')
      CALL PLOTR(WS,YS, )
      WRITE(6,101)
101   FORMAT(1H1,'PLOTTING OF THE FILTER PHASE RESPONSE')
      CALL PLOTR(WS,PHASES, )
      IF(L.EQ.0)GO TO 502
      DO 102 I=1,K
      II=I+I
      POLE..(I)=SNGL(..EAL(POLE1(I)))
      POLE.(II)=SNGL(REAL(POLE2(I)))
      POLEI(I)=SNGL(AIMAG(POLE1(I)))
      POLEI(II)=SNGL(AIMAG(POLE2(I)))
      ZERO..(I)=SNGL(REAL(ZERO..(I)))
      ZERO.(II)=SNGL(REAL(ZERO2(I)))
      ZERO..(I)=SNGL(AIMAG(ZERO1(I)))
102   ZE..I(II)=SNGL(AIMAG(ZERO2(I)))
502   I=1
      IF(L.EQ.0)GO TO 503
      LL=2K+1
      L2=LL+L-1
      DO 103 II=LL,L2
      POLE.(II)=SNGL(..(I))
      POLEI(II)=.0
      ZERO.(II)=SNGL(..(I))
      ZER.I(II)=.0
103   I=I+1
503   ..=2K+L
      CALL ZPLOT(POLER,POLEI,ZERO.,ZEROI,..)
      ....(6,400)(YS(I),YDS(I),PHASES(I),WS(I),I=1,.)
400   FORMAT(..12.5)
      M=M+1
      ..(.)=.(.)
      .(.)=.(4)
      .(.)=.(6)
      .(.)=.(.)
      .(.)=.(2)
      .(.)=.(.)
      .(.)=.(.)
      IF(M.LT..2)GO TO 1
      STOP
      END
```

```
      $ENTRY

C     ....MAIN PROGRAM [illegible]
C     [illegible]
      DOUBLE PRECISION X( ), [illegible]
     1 S1,S2,S3,S4,PL(30), [illegible]
      COMPLEX*16 Z(30), [illegible] (3,30), [illegible] (3,30), [illegible] (2,30), [illegible] (2,30)
C
C
C.....DIMENSION    Y0( ), [illegible] ( ),F( ), [illegible] L( ),Z( ), [illegible] ( ), [illegible] ( , ), [illegible] ( , ), [illegible] (L,
C
      COMMON /C1/Z,
      COMMON /C2/Y0,
      COMMON /C3/ ,K,L
      [illegible]=0.D0
      SUM1=0.D0
      SUM2=0.D0
      DO 2 J=1,
      [illegible]=DCMPLX(1. [illegible])
      IF( [illegible] ) GO TO 10
      DO 1 I=1,
      [illegible]
      [illegible]
      [illegible]
      [illegible] (I,J)=Z(J) [illegible]
      [illegible] (I,J)=Z(J) [illegible]
    1 [illegible]
      [illegible] (I,J)/ [illegible] (I,J))
   10 I=
      [illegible]
      [illegible]
      L [illegible] =LL+L-1
      [illegible] =LL,L [illegible]
      [illegible]
      [illegible]
      [illegible] (I,J)=Z(J) [illegible]
      [illegible] (I,J)=Z(J) [illegible]
      [illegible] (I,J)/ [illegible] (I,J))
    2 I= [illegible]
      [illegible]
      [illegible] (J)= [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible] J=1,
      [illegible] (J)= [illegible] (J)- [illegible] (J)
      [illegible]
      IF( [illegible] )
      [illegible] I=1,
      [illegible]
      [illegible]
      [illegible]
      S1=0.D0
      S2=0.D0
      S3=0.D0
      S4=0.D0
      DO [illegible] J=1,
      PL(J)=2. [illegible] (J) [illegible] (J)
      D1=DSIN(X(I))
      D2=DCOS(X(I))
      D3=DSIN(X(K1))
```

```
      B4=DCOS(X(K1))
      [illegible]=DSIN(X(K2))
      B6=DCOS(X(K2))
      B7=DSIN(X(K3))
      B8=DCOS(X(K3))
      A1=(-4.*B1*[illegible]*Z(J)+4.*(B1**3)*B2)/[illegible](I,J)
      A2=([illegible].*([illegible])/[illegible](I,J)
      A3=(-4.*B5*B6*[illegible]*Z(J)+4.*(B5**3)*B6)/[illegible]2(I,J)
      A4=(2.*(B5**2)*B7)/D2(I,J)
      S1=S1+PL(J)*[illegible]L(A1)
      S2=S2+PL(J)*[illegible]L(A2)
      S3=S3-PL(J)*[illegible]L(A3)
      S4=S4-PL(J)*[illegible]L(A4)
      D[illegible](I)=S1
      D[illegible](K1)=S2
      D[illegible](K2)=S3
    7 D[illegible](K3)=S4
   17 I=I
      IF(L.EQ.0)GO TO 1[illegible]
      LL=4*K+1
      L2=LL+L-1
      DO [illegible] II=LL,L2
      K4=II
      K5=II+L
      S1=0.[illegible]
      S2=[illegible].
      D[illegible] [illegible] J=1[illegible]
      [illegible]L(J)=2.[illegible](J)[illegible](J)
      [illegible]1=[illegible]S(X([illegible]))/[illegible](I,J)
      [illegible]2=[illegible]S(X(K5))/[illegible](I,J)
      S1=S1-[illegible]L(J)[illegible]L([illegible])
   1[illegible] S2=S2+PL(J)*[illegible]L([illegible]2)
      [illegible](K4)=S1
      [illegible](K5)=S2
    [illegible] I=I+1
   1[illegible] CONTINUE
      RETURN
      [illegible]
```

```
      DOUBLE PRECISION FUNCTION DREAL(X)
      COMPLEX*16 X,DCONJG
      DREAL=(X+DCONJG(X))/2.0
      RETURN
      END
```

```
      $ENTRY


      DOUBLE PRECISION FUNCTION DIMAG(X)
      COMPLEX*16 P,X,DCMPLX,DCONJG
      P=DCMPLX(0.D0,1.D0)
      DIMAG=(X-DCONJG(X))/(2.D0*P)
      RETURN
      END
```

```
      SUBROUTINE ZPLOT(POLER,POLEI,ZEROR,ZEROI,N)
      LOGICAL*1 LINE,DOT/'.'/,SPACE/' '/,CROSS/'X'/,NAUGHT/'O'/,
     1DASH/'-'/,STROKE/'I'/
      DIMENSION LINE(99),POLER(1),ZEROR(1),POLEI(1),ZEROI(1)
      PRINT 500
      DO 2 J=1,59
      DO 3 I=1,99
    3 LINE(I)=SPACE
      LINE(50)=STROKE
      IF(J-30)4,5,6
    4 Y=30-J
   80 IF(Y.NE.29.)GO TO 8
      NX=0
      GO TO 9
    8 NX=(Y/(TAN(ARSIN(Y/29.))*0.6))+1.
    9 LINE(50-NX)=DOT
      LINE(50+NX)=DOT
      GO TO 10
    5 DO 7 I=1,99
    7 LINE(I)=DASH
      GO TO 10
    6 Y=J-30
      GO TO 80
   10 CONTINUE
      DO 20 I=1,N
      NPOLEI=(1.-POLEI(I))*29.+1.
      IF(J.NE.NPOLEI)GO TO 25
      NPOLER=(1.+POLER(I))*49.+1.
      LINE(NPOLER)=CROSS
   25 CONTINUE
      NZEROI=(1.-ZEROI(I))*29.+1.
      IF(J.NE.NZEROI)GO TO 26
      NZEROR=(1.+ZEROR(I))*49.+1.
      LINE(NZEROR)=NAUGHT
   26 CONTINUE
   20 CONTINUE
      PRINT 60,LINE
    2 CONTINUE
   60 FORMAT(20X,101A1)
  500 FORMAT(1H1)
      RETURN
      END
```

```
C
C     ..................................................................
C
C        SUBROUTINE DFMFP
C
C        PURPOSE
C           TO FIND A LOCAL MINIMUM OF A FUNCTION OF SEVERAL VARIABLES
C           BY THE METHOD OF FLETCHER AND POWELL
C
C        USAGE
C           CALL DFMFP(FUNCT,N,X,F,G,EST,EPS,LIMIT,IER,H)
C
C        DESCRIPTION OF PARAMETERS
C           FUNCT  - USER-WRITTEN SUBROUTINE CONCERNING THE FUNCTION TO
C                    BE MINIMIZED. IT MUST BE OF THE FORM
C                    SUBROUTINE FUNCT(N,ARG,VAL,GRAD)
C                    AND MUST SERVE THE FOLLOWING PURPOSE
C                    FOR EACH N-DIMENSIONAL ARGUMENT VECTOR  ARG,
C                    FUNCTION VALUE AND GRADIENT VECTOR MUST BE COMPUTED
C                    AND, ON RETURN, STORED IN VAL AND GRAD RESPECTIVELY
C                    ARG,VAL AND GRAD MUST BE OF DOUBLE PRECISION.
C           N      - NUMBER OF VARIABLES
C           X      - VECTOR OF DIMENSION N CONTAINING THE INITIAL
C                    ARGUMENT WHERE THE ITERATION STARTS. ON RETURN,
C                    X HOLDS THE ARGUMENT CORRESPONDING TO THE
C                    COMPUTED MINIMUM FUNCTION VALUE
C                    DOUBLE PRECISION VECTOR.
C           F      - SINGLE VARIABLE CONTAINING THE MINIMUM FUNCTION
C                    VALUE ON RETURN, I.E. F=F(X).
C                    DOUBLE PRECISION VARIABLE.
C           G      - VECTOR OF DIMENSION N CONTAINING THE GRADIENT
C                    VECTOR CORRESPONDING TO THE MINIMUM ON RETURN,
C                    I.E. G=G(X).
C                    DOUBLE PRECISION VECTOR.
C           EST    - IS AN ESTIMATE OF THE MINIMUM FUNCTION VALUE.
C                    SINGLE PRECISION VARIABLE.
C           EPS    - TESTVALUE REPRESENTING THE EXPECTED ABSOLUTE ERROR.
C                    A REASONABLE CHOICE IS 10**(-16), I.E.
C                    SOMEWHAT GREATER THAN 10**(-D), WHERE D IS THE
C                    NUMBER OF SIGNIFICANT DIGITS IN FLOATING POINT
C                    REPRESENTATION.
C                    SINGLE PRECISION VARIABLE.
C           LIMIT  - MAXIMUM NUMBER OF ITERATIONS.
C           IER    - ERROR PARAMETER
C                    IER = 0 MEANS CONVERGENCE WAS OBTAINED
C                    IER = 1 MEANS NO CONVERGENCE IN LIMIT ITERATIONS
C                    IER =-1 MEANS ERRORS IN GRADIENT CALCULATION
C                    IER = 2 MEANS LINEAR SEARCH TECHNIQUE INDICATES
C                    IT IS LIKELY THAT THERE EXISTS NO MINIMUM.
C           H      - WORKING STORAGE OF DIMENSION N*(N+7)/2.
C                    DOUBLE PRECISION ARRAY.
C
C        REMARKS
C         I) THE SUBROUTINE NAME REPLACING THE DUMMY ARGUMENT  FUNCT
C            MUST BE DECLARED AS EXTERNAL IN THE CALLING PROGRAM.
C        II) IER IS SET TO 2 IF , STEPPING IN ONE OF THE COMPUTED
C            DIRECTIONS, THE FUNCTION WILL NEVER INCREASE WITHIN
C            A TOLERABLE RANGE OF ARGUMENT.
```

```
      C        IER = 2 MAY OCCUR ALSO IF THE INTERVAL WHERE F
      C        INCREASES IS SMALL AND THE INITIAL ARGUMENT WAS
      C        RELATIVELY FAR AWAY FROM THE MINIMUM SUCH THAT THE
      C        MINIMUM WAS OVERLEAPED. THIS IS DUE TO THE SEARCH
      C        TECHNIQUE WHICH DOUBLES THE STEPSIZE UNTIL A POINT
      C        IS FOUND WHERE THE FUNCTION INCREASES.
      C
      C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
      C        FUNCT
      C
      C     METHOD
      C        THE METHOD IS DESCRIBED IN THE FOLLOWING ARTICLE
      C        R. FLETCHER AND M.J.D. POWELL, A RAPID DESCENT METHOD FOR
      C        MINIMIZATION,
      C        COMPUTER JOURNAL VOL.6, ISS. 2, 1963, PP.163-168.
      C
      C     ..................................................................
      C
0001        SUBROUTINE DFMFP(FUNCT,N,X,F,G,EST,EPS,LIMIT,IER,H)
      C
      C        DIMENSIONED DUMMY VARIABLES
0002        DIMENSION H(1),X(1),G(1)
0003        DOUBLE PRECISION X,F,FX,FY,OLDF,HNRM,GNRM,H,G,DX,DY,ALFA,DALFA,
           1AMBDA,T,Z,W
      C
      C        COMPUTE FUNCTION VALUE AND GRADIENT VECTOR FOR INITIAL ARGUMENT
0004        CALL FUNCT(N,X,F,G)
      C
      C        RESET ITERATION COUNTER AND GENERATE IDENTITY MATRIX
0005        IER=0
0006        KOUNT=0
0007        N2=N+N
0008        N3=N2+N
0009        N31=N3+1
0010      1 K=N31
0011        DO 4 J=1,N
0012        H(K)=1.D0
0013        NJ=N-J
0014        IF(NJ)5,5,2
0015      2 DO 3 L=1,NJ
0016        KL=K+L
0017      3 H(KL)=0.D0
0018      4 K=KL+1
      C
      C        START ITERATION LOOP
0019      5 KOUNT=KOUNT +1
      C
      C        SAVE FUNCTION VALUE, ARGUMENT VECTOR AND GRADIENT VECTOR
0020        OLDF=F
0021        DO 9 J=1,N
0022        K=N+J
0023        H(K)=G(J)
0024        K=K+N
0025        H(K)=X(J)
      C
      C        DETERMINE DIRECTION VECTOR H
0026        K=J+N3
0027        T=0.D0
```

```
      DO 8 L=1,N
      T=T-G(L)*H(K)
      IF(L-J)6,7,7
    6 K=K+N-L
      GO TO 8
    7 K=K+1
    8 CONTINUE
    9 H(J)=T
C
C        CHECK WHETHER FUNCTION WILL DECREASE STEPPING ALONG H.
      DY=0.D0
      HNRM=0.D0
      GNRM=0.D0
C
C        CALCULATE DIRECTIONAL DERIVATIVE AND TESTVALUES FOR DIRECTION
C        VECTOR H AND GRADIENT VECTOR G.
      DO 10 J=1,N
      HNRM=HNRM+DABS(H(J))
      GNRM=GNRM+DABS(G(J))
   10 DY=DY+H(J)*G(J)
C
C        REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF DIRECTIONAL
C        DERIVATIVE APPEARS TO BE POSITIVE OR ZERO.
      IF(DY)11,51,51
C
C        REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF DIRECTION
C        VECTOR H IS SMALL COMPARED TO GRADIENT VECTOR G.
   11 IF(HNRM/GNRM-EPS)51,51,12
C
C        SEARCH MINIMUM ALONG DIRECTION H
C
C        SEARCH ALONG H FOR POSITIVE DIRECTIONAL DERIVATIVE
   12 FY=F
      ALFA=2.D0*(EST-F)/DY
      AMBDA=1.D0
C
C        USE ESTIMATE FOR STEPSIZE ONLY IF IT IS POSITIVE AND LESS THAN
C        1. OTHERWISE TAKE 1. AS STEPSIZE
      IF(ALFA)15,15,13
   13 IF(ALFA-AMBDA)14,15,15
   14 AMBDA=ALFA
   15 ALFA=0.D0
C
C        SAVE FUNCTION AND DERIVATIVE VALUES FOR OLD ARGUMENT
   16 FX=FY
      DX=DY
C
C        STEP ARGUMENT ALONG H
      DO 17 I=1,N
   17 X(I)=X(I)+AMBDA*H(I)
C
C        COMPUTE FUNCTION VALUE AND GRADIENT FOR NEW ARGUMENT
      CALL FUNCT(N,X,F,G)
      FY=F
C
C        COMPUTE DIRECTIONAL DERIVATIVE DY FOR NEW ARGUMENT.  TERMINATE
C        SEARCH, IF DY IS POSITIVE. IF DY IS ZERO THE MINIMUM IS FOUND
      DY=0.D0
```

```
      DO 18 I=1,N
   18 DY=DY+G(I)*H(I)
      IF(DY)19,36,22
C
C     TERMINATE SEARCH ALSO IF THE FUNCTION VALUE INDICATES THAT
C     A MINIMUM HAS BEEN PASSED
   19 IF(FY-FX)20,22,22
C
C     REPEAT SEARCH AND DOUBLE STEPSIZE FOR FURTHER SEARCHES
   20 AMBDA=AMBDA+ALFA
      ALFA=AMBDA
C     END OF SEARCH LOOP
C
C     TERMINATE IF THE CHANGE IN ARGUMENT GETS VERY LARGE
      IF(HNRM*AMBDA-1.D10)16,16,21
C
C     LINEAR SEARCH TECHNIQUE INDICATES THAT NO MINIMUM EXISTS
   21 IER=2
      RETURN
C
C     INTERPOLATE CUBICALLY IN THE INTERVAL DEFINED BY THE SEARCH
C     ABOVE AND COMPUTE THE ARGUMENT X FOR WHICH THE INTERPOLATION
C     POLYNOMIAL IS MINIMIZED
   22 T=0.D0
   23 IF(AMBDA)24,36,24
   24 Z=3.D0*(FX-FY)/AMBDA+DX+DY
      ALFA=DMAX1(DABS(Z),DABS(DX),DABS(DY))
      DALFA=Z/ALFA
      DALFA=DALFA*DALFA-DX/ALFA*DY/ALFA
      IF(DALFA)51,25,25
   25 W=ALFA*DSQRT(DALFA)
      ALFA=DY-DX+W+W
      IF(ALFA) 250,251,250
  250 ALFA=(DY-Z+W)/ALFA
      GO TO 252
  251 ALFA=(Z+DY-W)/(Z+DX+Z+DY)
  252 ALFA=ALFA*AMBDA
      DO 26 I=1,N
   26 X(I)=X(I)+(T-ALFA)*H(I)
C
C     TERMINATE, IF THE VALUE OF THE ACTUAL FUNCTION AT X IS LESS
C     THAN THE FUNCTION VALUES AT THE INTERVAL ENDS. OTHERWISE REDUCE
C     THE INTERVAL BY CHOOSING ONE END-POINT EQUAL TO X AND REPEAT
C     THE INTERPOLATION.  WHICH END-POINT IS CHOOSEN DEPENDS ON THE
C     VALUE OF THE FUNCTION AND ITS GRADIENT AT X
C
      CALL FUNCT(N,X,F,G)
      IF(F-FX)27,27,28
   27 IF(F-FY)36,36,28
   28 DALFA=0.D0
      DO 29 I=1,N
   29 DALFA=DALFA+G(I)*H(I)
      IF(DALFA)30,33,33
   30 IF(F-FX)32,31,33
   31 IF(DX-DALFA)32,36,32
   32 FX=F
      DX=DALFA
      T=ALFA
```

```
0096             AMBDA=ALFA
0097             GO TO 23
0098          33 IF(FY-F)35,34,35
0099          34 IF(DY-DALFA)35,36,35
0100          35 Y=F
0101             DY=DALFA
0102             AMBDA=AMBDA-ALFA
0103             GO TO 22
           C
           C     TERMINATE, IF FUNCTION HAS NOT DECREASED DURING LAST ITERATION
0104          36 IF(OLDF-F+EPS)51,38,38
           C
           C     COMPUTE DIFFERENCE VECTORS OF ARGUMENT AND GRADIENT FROM
           C     TWO CONSECUTIVE ITERATIONS
0105          38 DO 37 J=1,N
0106             K=N+J
0107             H(K)=G(J)-H(K)
0108             K=N+K
0109          37 H(K)=X(J)-H(K)
           C
           C     TEST LENGTH OF ARGUMENT DIFFERENCE VECTOR AND DIRECTION VECTOR
           C     IF AT LEAST N ITERATIONS HAVE BEEN EXECUTED. TERMINATE, IF
           C     BOTH ARE LESS THAN   EPS
0110             IER=0
0111             IF(KOUNT-N)42,39,39
0112          39 T=0.D0
0113             Z=0.D0
0114             DO 40 J=1,N
0115             K=N+J
0116             W=H(K)
0117             K=K+N
0118             T=T+DABS(H(K))
0119          40 Z=Z+W*H(K)
0120             IF(HNRM-EPS)41,41,42
0121          41 IF(T-EPS)56,56,42
           C
           C     TERMINATE, IF NUMBER OF ITERATIONS WOULD EXCEED  LIMIT
0122          42 IF(KOUNT-LIMIT)43,50,50
           C
           C     PREPARE UPDATING OF MATRIX H
0123          43 ALFA=0.D0
0124             DO 47 J=1,N
0125             K=J+N3
0126             W=0.D0
0127             DO 46 L=1,N
0128             KL=N+L
0129             W=W+H(KL)*H(K)
0130             IF(L-J)44,45,45
0131          44 K=K+N-L
0132             GO TO 46
0133          45 K=K+1
0134          46 CONTINUE
0135             K=N+J
0136             ALFA=ALFA+W*H(K)
0137          47 H(J)=W
           C
           C     REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF RESULTS
           C     ARE NOT SATISFACTORY
```

```
      IF(Z*ALFA)48,1,48
C
C     UPDATE MATRIX H
   48 K=N21
      DO 49 L=1,N
      KL=N2+L
      DO 49 J=L,N
      NJ=N2+J
      H(K)=H(K)+H(KL)*H(NJ)/Z-H(L)*H(J)/ALFA
   49 K=K+1
      GO TO 5
C     END OF ITERATION LOOP
C
C     NO CONVERGENCE AFTER LIMIT ITERATIONS
   50 IER=1
      RETURN
C
C     RESTORE OLD VALUES OF FUNCTION AND ARGUMENTS
   51 DO 52 J=1,N
      K=N2+J
   52 X(J)=H(K)
      CALL FUNCT(N,X,F,G)
C
C     REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF DERIVATIVE
C     FAILS TO BE SUFFICIENTLY SMALL
      IF(GNRM-EPS)55,55,53
C
C     TEST FOR REPEATED FAILURE OF ITERATION
   53 IF(IER)56,54,56
   54 IER=-1
      GOTO 1
   55 IER=0
   56 RETURN
      END
```

```
$ENTRY

C..DESIGN OF A DIGITAL FILTER HAVING A REQUIRED MAGNITUDE AND PHASE
C   BY MINIMIZING A SINGLE ERROR CRITERION
C   MAIN PROGRAM
      DIMENSION POLER(8),POLEI(8),ZEROR(8),ZEROI(8),YS(30),YDS(30),
     1 WS(30),PHASDS(30),PHASES(30)
      DOUBLE PRECISION X(7 ),FD(7 ),YD(30),W(30),HC(49 ),D,A(2),B(2),
     1 C(2),D(2),AJ(1),BJ(1),PY,PHASE(30),PHASD(30),Y(30),DSIN,E(2),
     1 ,E1(30),E2(30),D1,D2,DATAN,CDABS
      COMPLEX*16 Z(30),H(30),CD,P1,P2,CDEXP,POLE1(2) ,POLE2(2) ,
     1 P,ZERO1(2) ,ZERO2(2) ,DCONJG,DCMPLX
      COMMON /B1/Z,H
      COMMON /B2/PHASE,PHASD,YD,W,PY,E1,E2
      COMMON /B3/ALAMD1,ALAMD2
      COMMON /B4/M,K,L
      EXTERNAL FUNCT
      PY=3.14159102D00
C ....THE FILTER SPECIFICATIONS ARE WRITTEN BETWEEN THE DOTTED LINES....
C ......................................................................
C   ,M,KL   [illegible] SPECIFIED IN THE LITERATURE
C.... [illegible]
C ..DIMENSION ..X(M).. [illegible] IS THE NUMBER OF VARIABLES [illegible] ,KL
C.. YD( ).[illegible] ( ),W( ).[illegible] ( ),D( ).C( ).D( ).E( ).AJ(L).BJ(L),Y( ),
C  ,Z( ),H( ),POLE1( ).POLE2( ),ZERO1( ).ZERO2( )
C..DIMENSION [illegible] POLER( ),POLEI( ) ZEROR( ),ZEROI( ),...
C

C...YD(I) & PHASD(I) ARE THE MAGNITUDE & PHASE DESIRED AT THE [illegible]
C..W(I) IS THE [illegible]
C.. [illegible]
C  LIMIT IS THE [illegible] OF FUNCTION EVALUATION REQUIRED
C
C  LAMD1 ..IS THE WEIGHTING FACTOR OF THE ERROR IN MAGNITUDE, AND ALAMD2
C  ..IS THE WEIGHTING FACTOR OF THE ERROR IN PHASE
C
C..THE VECTOR X STORES THE VARIABLES CONSECUTIVELY,I.E.,THE FIRST [illegible]
C.SAY,M,IS STORED FROM 1 TO M,THE SECOND VARIABLE ,SAY [illegible]
      ALAMD1=1.[illegible]
      ALAMD2=[illegible]
      M=21
      K=1
      L=1
C..D(1)=X(1).C(1)=X(2),A (1)=X(3),B D(1)=X(4),AJ(1)=X(5),[illegible]
C...[illegible] K=2,[illegible] A(1)=X(1),A(2)=X(2),...ETC.
      X(1)=-0.[illegible]
      X(2)=-0.1[illegible]477[illegible]D-07
      X(3)=-0.2[illegible]167D-01
      X(4)=-0.[illegible]5241[illegible]D-02
      X(5)=0.205[illegible]5752[illegible]D 00
      X(6)=0.[illegible]270[illegible]77[illegible]25D 00
      X(7)=-0.1[illegible]
      PHASD(1)=0.D0
      W(1)=0.D0
      YD(1)=0.D0
      DO 3 I=1,20
      W(I+1)=W(I)+0.05
      PHASD(I+1)=W(I+1)*PY
    3 YD(I+1)=W(I+1)
      LIMIT=150
C ......................................................................
```

```
      DO 4 I=1,N
      CP=DCMPLX(0.D0,W(I)*PY)
    4 Z(I)=CDEXP(CP)
      EST=0.1
      EPS=1.0/10.**5
      CALL DEFER(FUNCT,N,X,0.D0,EST,EPS,LIMIT,IER,WC)
      WRITE(6,5)IER
    5 FORMAT(5X,'IER=',I3)
      WRITE(6,7)D
    7 FORMAT(5X,'D=',D20.12)
      WRITE(6,502) (I,X(I),I=1,N)
  502 FORMAT(5X,'X(',I3,')=',D20.12)
      DO 14 I=1,K
      K1=K+I
      K2=2*K+I
      K3=3*K+I
      K4=4*K+I
      A(I)=X(I)
      B(I)=X(K1)
      C(I)=X(K2)
      D(I)=-(2.*DSIN(X(K3))**2)*DCOS(X(K4))
      E(I)=DSIN(X(K3))**4
      WRITE(6,9)(I,A(I),I,B(I),I,C(I),I,D(I),I,E(I))
    9 FORMAT(5X,'A(',I3,')=',D20.12/5X,'B(',I3,')=',D20.12/5X,'C(I',I3,
     1 ')=',D20.12/5X,'D(',I3,')=',D20.12/5X,'E(',I3,')=',D20.12)
      Z1=B(I)**2-4.*A(I)*C(I)
      Z2=D(I)**2-4.*E(I)
      POLE1(I)=(-B(I)-CDSQRT(Z1))/2.[illegible]
      POLE2(I)=(-B(I)+CDSQRT(Z1))/2.
      ZERO1(I)=-(D(I)+CDSQRT(Z2))/(2.*E(I))
      ZERO2(I)=-(D(I)-CDSQRT(Z2))/(2.*E(I))
      WRITE(6,15)(I,POLE1(I))
      WRITE(6,16)(I,POLE2(I))
      WRITE(6,17)(I,ZERO1(I))
      WRITE(6,18)(I,ZERO2(I))
   15 FORMAT(5X,'POLE1(',I3,')=',D20.12,3X,D20.12)
   16 FORMAT(5X,'POLE2(',I3,')=',D20.12,3X,D20.12)
   17 FORMAT(5X,'ZERO1(',I3,')=',D20.12,3X,D20.12)
   18 FORMAT(5X,'ZERO2(',I3,')=',D20.12,3X,D20.12)
   14 CONTINUE
      LL=5K+1
      J=1
      L2=LL+L-1
      DO 10 JJ=LL,L2
      K6=JJ
      K7=JJ+L
      TJ(J)=X(K6)
      WJ(J)=[illegible](X(K7))
      WRITE(6,20)(J,TJ(J))
      WRITE(6,21)(J,WJ(J))
   20 FORMAT(5X,'TJ(',I3,')=',D20.12)
   21 FORMAT(5X,'WJ(',I3,')=',D20.12)
   10 J=J+1
      DO 30 I=1,N
      [illegible](I)=[illegible]SQ(I)*1[illegible].00/[illegible]
   30 PHASD(I)=PHASD(I)*1[illegible].00/PY
      DO 34 I=1,
      [illegible]S(I)=SNGL([illegible](I))
      PHASDS(I)=SNGL(PHASD(I))
   34 YOS(I)=SNGL(YO(I))
      WRITE(6,24)
```

```
24  FORMAT(1H1,'PLOTTING OF THE GIVEN FILTER MAGNITUDE RESPONSE')
    CALL PLOT3(WS,YDS,N)
    WRITE (6,28)
28  FORMAT(1H1,'PLTTING OF THE GIVEN FILTER PHASE RESPONSE')
    CALL PLOT3(WS,PHASES,N)
    DO 22 I=1,N
22  Y(I)=C[illegible](I))
    WRITE(6,72)
72  FORMAT(4X,'I',20X,'Y(I)',30X,'W(I)',30X,'PHASE(I)')
    WRITE(6,73)(I,Y(I),W(I),PHASE(I),I=1,N)
73  FORMAT(4X,I3,6X,D20.12,21X,D20.12,20X,D20.12)
    DO 21 I=1,N
    PHASES(I)=SNGL(PHASE(I))
21  YS(I)=SNGL(Y(I))
    WRITE(6,26)
26  FORMAT(1H1,'PLOTTING OF THE DESIGNED FILTER MAGNITUDE RESPONSE')
    CALL PLOT3(WS,YS,N)
    WRITE(6,29)
29  FORMAT(1H1,'PLOTTING OF THE DESIGNED FILTER PHASE RESPONSE')
    CALL PLOT3(WS,PHASES,N)
    DO 102 I=1,K
    KI=K+I
    POLER(I)=SNGL(DREAL(POLE1(I)))
    POLER(KI)=SNGL(DREAL(POLE2(I)))
    POLEI(I)=SNGL(DIMAG(POLE1(I)))
    POLEI(KI)=SNGL(DIMAG(POLE2(I)))
    ZEROR(I)=SNGL(DREAL(ZERO1(I)))
    ZEROR(KI)=SNGL(DREAL(ZERO2(I)))
    ZEROI(I)=SNGL(DIMAG(ZERO1(I)))
102 ZEROI(KI)=SNGL(DIMAG(ZERO2(I)))
    I=1
    LL=2*K+1
    L2=LL+L-1
    DO 103 II=LL,L2
    POLER(II)=SNGL(PJ(I))
    POLEI(II)=0.0
    ZEROR(II)=SNGL(AJ(I))
    ZEROI(II)=0.0
103 I=I+1
    NP=2*K+L
    CALL ZPLOT(POLER,POLEI,ZEROR,ZEROI,NP)
    Q1=0.D0
    Q2=0.D0
    DO 500 I=1,N
    Q1=Q1+F1(I)**2
500 Q2=Q2+F2(I)**2
    WRITE(6,600) Q1,Q2
600 FORMAT(3X,'SQUARE ERROR IN MAGNITUDE=',D20.12/3X,'SQUARE ERROR
   1 IN PHASE=',D20.12)
    PUNCH 4000,(YS(I),WS(I),PHASES(I),I=1,N)
4000 FORMAT(3F12.5)
    STOP
    END
```

SENTRY

```
      SUBROUTINE FUNCT( ,X, , )
C
C
C...DIMENSIONING..PHASE( ),PHASD( ),YD( ), ( ),E1( ),...............
C..E2( ),PL1( ),PL2( ),C( ),AI( ),B( ),Z( ),H( ),R1( , ),R2( , ),R7( ,
C
C
      DOUBLE PRECISION X( ),Q,DQ( ),PHASE(30),PHASD(30),YD(30), (30),
     1PY,E1(30),E2(30),S1,S2,S3,S4,S5,PL1(30),PL2(30),AI(30),C(30), (3
     1 ,A6,A7,A8,A9,A10,ARG,DREAL,DIMAG,DSIN,DCOS,DATAN,DCOS
      COMPLEX*16 Z(30),H(30),R1(2,30),R2(2,30),R,R7(1,30),R8(1,30),
     1 A1,A2,A3,A4,A5,DCMPLX
      COMMON /C1/Z,H
      COMMON/C2/PHASE,PHASD,YD, ,PY,E1,E2
      COMMON/C3/ALAMD1,ALAMD2
      COMMON /C4/ , ,L
      Q=0.D0
      DO 2 J=1,
      R=DCMPLX(1.D0,0.D0)
      DO 1 I=1,
      K1=K*I
      K2=2*K+I
      K3=3*K+I
      K4=4*K+I
      R1(I,J)=X(I)*(Z(J)**2)+X( )*Z(J)+X( )
      R2(I,J)=Z(J)**2-2.*(DSIN(X(K3))**2)*DCOS(X(K4))*Z(J)+DSIN(X(
     1 K4)
    1 R=R*(R1(I,J)/R2(I,J))
      I=1
      LL=5*K+1
      L2=LL+L-1
      DO 3 II=LL,L2
      K5=II
      K6=II+L
      R7(I,J)=Z(J)-X(K5)
      R8(I,J)=Z(J)-DSIN(X(K6))
      R=R*(R7(I,J)/R8(I,J))
    3 I=I+1
      H(J)=R
      ARG=DIMAG(H(J))/DREAL(H(J))
      PHASE(J)=DATAN(ARG)
      E1(J)=DCOS(H(J))-YD(J)
      E2(J)=PHASE(J)-PHASD(J)
    2 Q=Q+(ALAMD1*E1(J)**2)+(ALAMD2*E2(J)**2)
      DO 7 I=1,K
      K1=K+I
      K2=2*K+I
      K3=3*K+I
      K4=4*K+I
      S1=0.D0
      S2=0.D0
      S3=0.D0
      S4=0.D0
      S5=0.D0
      DO 8 J=1,
      PL1(J)=2.0*ALAMD1*E1(J)*CDABS(H(J))
      PL2(J)=2.0*ALAMD2*E2(J)
      C(J)=CDABS(H(J))**2.
      B(J)=DREAL(H(J))
```

```
      A1=Z(J)**2/R1(I,J)
      A2=Z(J)/R1(I,J)
      A3=1.0/R1(I,J)
      A4=(4.0*DSIN(X(K3))*DCOS(X(K3))*DCOS(X(K4))*Z(J)-4.0*(DSIN(X(K3
     1 **3)*DCOS(X(K3)))/R2(I,J)
      A5=-2.0*(DSIN(X(K3))**2)*DSIN(X(K4))*Z(J)/R2(I,J)
      A6=(B(J)*DIMAG(A1*H(J))-AI(J)*DREAL(A1*H(J)))/C(J)
      A7=(B(J)*DIMAG(A2*H(J))-AI(J)*DREAL(A2*H(J)))/C(J)
      A8=(B(J)*DIMAG(A3*H(J))-AI(J)*DREAL(A3*H(J)))/C(J)
      A9=(B(J)*DIMAG(A4*H(J))-AI(J)*DREAL(A4*H(J)))/C(J)
      A10=(B(J)*DIMAG(A5*H(J))-AI(J)*DREAL(A5*H(J)))/C(J)
      S1=S1+PL1(J)*DREAL(A1)+PL2(J)*A6
      S2=S2+PL1(J)*DREAL(A2)+PL2(J)*A7
      S3=S3+PL1(J)*DREAL(A3)+PL2(J)*A8
      S4=S4+PL1(J)*DREAL(A4)+PL2(J)*A9
    8 S5=S5+PL1(J)*DREAL(A5)+PL2(J)*A10
      DO(I)=S1
      DO(K1)=S2
      DO(K2)=S3
      DO(K3)=S4
    7 DO(K4)=S5
      I=1
      LL=5*K+1
      L2=LL+L-1
      DO 10 II=LL,L2
      K5=II
      K6=II+L
      S1=0.D0
      S2=0.D0
      DO 9 J=1,N
      A1=-1.0/R2(I,J)
      A2=DCOS(X(K6))/R2(I,J)
      A3=(B(J)*DIMAG(A1*H(J))-AI(J)*DREAL(A1*H(J)))/C(J)
      A4=(B(J)*DIMAG(A2*H(J))-AI(J)*DREAL(A2*H(J)))/C(J)
      S1=S1+PL1(J)*DREAL(A1)+PL2(J)*A3
    9 S2=S2+PL1(J)*DREAL(A2)+PL2(J)*A4
      DO(K5)=S1
      DO(K6)=S2
   10 I=I+1
      RETURN
      END
```

$ENTRY

```
C..DESIGN OF AN ANALOG FILTER USING BILINEAR TRANSFORMATION OF A DIGIT[illegible]
C    .... [illegible]
      DIMENSION YS(30),WS(30),PHAS(30),YDS(30)
      DOUBLE PRECISION X(16),DX(16),YD(31),W(31),HC(184),C,A(3),B(3),
     1 C(3),        D(3),AJ(2),BJ(2),DSTE,PY,Y(31),DCOS,[illegible]S,[illegible]
     1 ,DREAL,DTAN,AIMAG,DATAN,FLOAT ,PHASE(31)
     1 ,S1,S2,SUM1,SUM2,[illegible]A(3),C[illegible](3),D[illegible](3),AJ[illegible](2),BJ[illegible](2),A[illegible](3),[illegible]
      COMPLEX*16 Z(31),CP,DCMPLX,CDEXP,H(31),P1,P2,POLE1(3),POLE2(3)
     1 ,ZERO1(3),ZERO2(3),F,DCONJG,CDSQRT ,S(31)
      COMMON/B1/Z,N
      COMMON/B2/YD,AD
      COMMON/B3/[illegible],K,L
      PY=3.14159265358979
      EXTERNAL FUNCT
C
C
C..DIMENSIONING  X([illegible]),DX([illegible]),YD( ),W([illegible]),HC([illegible]*([illegible]+7)/2),[illegible](K),[illegible](K),C(K),[illegible]
C   BJ(L),Y( ),Z( ),POLE1(K),POLE2(K),ZERO1(K),ZERO2(K)
C..DIMENSIONING  [illegible]A(K),[illegible](K),C[illegible](K),[illegible](K),AJ[illegible](L),BJ[illegible](L)
C
C..N,K,L ARE AS SPECIFIED IN THE LITERATURE,[illegible] IS THE [illegible]
C ...THE FILTER SPECIFICATIONS ARE T[illegible] GIVE[illegible] THE [illegible]
C
C.........................................................................
C YD  IS THE DESI[illegible]
C ...[illegible]
C..LIMIT [illegible]
      K=3
      L=2
C ALP[illegible]=X(1),[illegible]=X(2),[illegible]=X(3),[illegible]=X(4),[illegible]=X(5),[illegible]=X(6),
C NOTE [illegible]=2,[illegible] ALP[illegible](1)=X(1),ALP[illegible](2)=X(2),[illegible](1)=X(3)..ETC
      X(1)=-0.81[illegible]
      X(2)=-0.81[illegible]
      X(3)=-0.81[illegible]
      X(4)=[illegible].[illegible]
      X(5)=[illegible].[illegible]
      X(6)=[illegible].[illegible]
      X(7)=[illegible].[illegible]
      X(8)=[illegible].77[illegible]
      X(9)= .77[illegible]
      X(10)=1.1[illegible]
      X(11)=1.1[illegible]
      X(12)=1.[illegible]
      X(13)=[illegible]
      X(14)=1.[illegible]
      X(15)=0.1[illegible]
      X(16)=0.1[illegible]
C..N =[illegible] POINTS SELECTED [illegible] FREQ. AXIS
      N=21
      W(1)=0.D0
      YD(1)=0.D0
      DO 1 I=1,10
      W(I+1)=W(I)+1000.D0
    1 YD(I+1)=YD(I)
      W(11)=10000.D0
      YD(11)=0.5D0
      DO 2 I=12,21
      YD(I)=1.D0
    2 W(I)=W(I-1)+3000.D0
```

FRACT

```
C. CONDITIONS NEEDED AFTER THE SELECTED FREQUENCIES SHOULD BE DONE I
C. OF 1.0, BETWEEN 0.5 AND 1.0, WHERE W(MU)=0.5, AND W(I)=1.0 IS IN S.
C.. FREQUENCY IN THE S-PLANE
C. DIGITAL FREQ. AXIS    0.0--------------0.5-------------1.0
C. ANALOG FREQ. AXIS     0.0--------------W(MU)-----------INFINITY
C*********************************************************************
C..
         .......
      ALAMD=1.00/W(MU)
C..SCALING FREQUENCIES AND WARPING
      DO 3 I=1,M
      W(I)=W(I)*ALAMD
    3 W(I)=2.0*DATAN(W(I))/PY
C*********************************************************************
      DO 5 I=M,29
      W(I+1)=W(I)+0.05
    5 YD(I+1)=1.00
      M=30
      LIMIT=30
C
C.....................................................................
C
      [illegible]
      DO 4 I=1,
      [illegible]
    4 Z(I)=[illegible]
      EST=[illegible]
      EPS=1.0/10.**[illegible]
      CALL [illegible],EST,EPS,LIMIT,[illegible]
      [illegible]
    [illegible]
      [illegible]
      [illegible]
      [illegible]
  100 [illegible]
      DO 12 I=1,[illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
   27 [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible](I)=-[illegible](I)/2.[illegible]
      [illegible](I)=-[illegible](I)/2.[illegible]/2.0
      [illegible](I)=-[illegible](I)/2.[illegible]/2.0
      [illegible](I)=-[illegible](I)/2.[illegible]/2.0
      [illegible]
      [illegible]
      [illegible]
      WRITE([illegible]) I,[illegible]Z(I)
   15 FORMAT([illegible],I2,[illegible],D20.12,3X,D20.12)
   16 FORMAT([illegible],I2,[illegible],D20.12,3X,D20.12)
   17 FORMAT(5X,[illegible],I2,[illegible],D20.12,3X,D20.12)
   18 FORMAT(5X,[illegible],I2,[illegible],D20.12,3X,D20.12)
```

```
   14 CONTINUE
      J=1
      LL=4*K+1
      L2=LL+L-1
      DO 19 JJ=LL,L2
      K3=JJ
      K4=JJ+L
      AJ(J)=DSIN(X(K3))
      BJ(J)=DSIN(X(K4))
      WRITE(6,20) J,AJ(J)
   20 FORMAT(5X,'AJ(',I3,')=',D20.12)
      WRITE(6,21) J,BJ(J)
   21 FORMAT(5X,'BJ(',I3,')=',D20.12)
   19 J=J+1
      DO 30 I=1,
      HS(I)=SNGL(H(I))
   30 YDS(I)=SNGL(YD(I))
      DO 25 I=1,
   25 Y(I)=[illegible]S([illegible](I))
      WRITE(6,71)
   71 FORMAT(4X,'I',20X,'Y(I)',20X,'H(I)')
      WRITE(6,72)(I,Y(I),H(I),I=1, )
   72 FORMAT(2X,I3,[illegible].12,21X,D20.12)
      DO 31 I=1,
   31 YS(I)=SNGL(Y(I))
      WRITE(6,24)
   24 FORMAT(1H1,'PLOTTING OF THE GIVEN FILTER RESPONSE')
      CALL [illegible]PTR( S,HS, )
      WRITE(6,26)
   26 FORMAT(1H1,'PLOTTING OF THE DESIGNED FILTER RESPONSE')
      CALL PLOTR( S,YS, )
      SUM1=0.D0
      SUM2=0.D0
      [illegible]=-1
      DO 102 J=1,
      SI=[illegible]([illegible](J)/[illegible].[illegible])
      S(J)=[illegible]([illegible])/[illegible]
      P=DCMPLX(1.D0,0.D0)
      DO 101 I=1,K
      [illegible](I)=(2.D0-2.D0*[illegible](I))/((1.D0-[illegible](I)*[illegible](I))*[illegible])
      [illegible](I)=([illegible](I)+[illegible](I))/((1.D0-[illegible](I)*[illegible](I))*[illegible])
      [illegible](I)=(2.D0-2.D0*[illegible](I))/((1.D0-[illegible](I)*[illegible](I))*[illegible])
      [illegible](I)=([illegible](I)+[illegible](I))/((1.D0-[illegible](I)*[illegible](I))*[illegible])
      [illegible]=S(J)*S(J)+[illegible](I)*S(J)+[illegible](I)
      P2=S(J)*S(J)+[illegible](I)*S(J)+[illegible](I)
  101 P=P*(P1/P2)
      DO 104 I=1,L
      [illegible](I)=(1.D0-[illegible](I))/((1.D0+[illegible](I))*[illegible])
      [illegible](I)=(1.D0-[illegible](I))/((1.D0+[illegible](I))*[illegible])
      P1=S(J)+[illegible](I)
      P2=S(J)+[illegible](I)
  104 P=P*(P1/P2)
      [illegible](J)=[illegible]
      SUM1=SUM1+CD[illegible]S([illegible](J))*[illegible](J)
  102 SUM2=SUM2+[illegible]S([illegible](J))**2
      C=SUM1/SUM2
      DO 105 I=1,
      YS(I)=SNGL([illegible]CDABS([illegible](I)))
  105 [illegible]S(I)=SNGL([illegible]T[illegible](PY*[illegible](I)/2.D0))/SNGL([illegible])
      WRITE(6,107)
  107 FORMAT(1H1,'[illegible] FILTER RESULTS')
```

```
      CALL PLOT3(XS,YS,N)
      WRITE(6,11) AAD
      DO 300 I=1,N
      PHAS (I)=DATAN(DIMAG(F(I))/DREAL(F(I)))
300   PHAS(I)=SNGL(PHASE(I))
      DO 106 I=1,K
      WRITE(6,22) I, (I),T.(I),T.C(I).I,(I)
      R1=DCMPLX(AA(I)**2-4.D0*B(I),0.D0)
      R2=DCMPLX(CA(I)**2-4.D0*D(I),0.D0)
      POLE1(I)=-CA(I)/2.D+CDSQRT(R1)/2.D
      POLE2(I)=-CA(I)/2.D-CDSQRT(R1)/2.D
      ZERO1(I)=-A(I)/2.D+CDSQRT(R2)/2.D
      ZERO2(I)=-AA(I)/2.D-CDSQRT(R2)/2.D
      WRITE(6.15) I,POLE1(I)
      WRITE(6,16) I,POLE2(I)
      WRITE(6,17) I,ZERO1(I)
106   WRITE(6.18) I,ZERO2(I)
      DO 108 I=1,L
      WRITE(6,20) I, AJA(I)
108   WRITE(6,21) I, BJA(I)
      WRITE(6,200)
200   FORMAT(4X,'I',20X,'Y(I)',30X,'X(I)')
      WRITE(6,201)(I,YS(I),XS(I),I=1,N)
201   FORMAT(2X.I3,10X,E12.5.10X.E12.5)
      PUNCH 400.(Y(I),X(I).PHASE(I),XP(I),I=1,N)
400   FORMAT(4E12.5)
      STOP
      END
```

SENTRY

```
C..DESIGN OF A DIGITAL FILTER USING STEIGLITZ'S METHOD
      DIMENSION YPS(30),[illegible]S(30),XS(50),[illegible]S(30),[illegible]L[illegible](4),[illegible]L[illegible]I(4),
     1 ZERO[illegible](4),ZER[illegible]I(4)
      DOUBLE PRECISION X([illegible]),[illegible]([illegible]),[illegible],[illegible],PY,YD(30),[illegible]([illegible]),[illegible]([illegible]),[illegible](2),C(
     1 [illegible](2),Y(32),[illegible]C(40),[illegible],[illegible]S[illegible](30),[illegible]L[illegible]
     1 ,Z1(2),Z2(2),[illegible](2),[illegible](2)
      COMPLEX*16 Z(30),CP,POLE1(2),POLE2(2),ZE[illegible]1(2),ZE[illegible]2(2),H(30)
     1 [illegible]S,[illegible],HC[illegible]L[illegible],[illegible] [illegible]
      EXTERNAL FUNCT
      CO[illegible] /[illegible]1/Z,[illegible]
      CO[illegible] /[illegible]2/Y[illegible],[illegible]
      CO[illegible] /[illegible]3/K,[illegible]
      CO[illegible] /[illegible]/[illegible]T
      PY=3.141592654
C..DIMENSIONING   X([illegible]),[illegible]([illegible]),YD([illegible]),[illegible]([illegible]),[illegible]([illegible]),[illegible]([illegible]),C([illegible]),[illegible]([illegible]),Y([illegible]),HC
C...Z([illegible]),POLE1([illegible]),POLE2([illegible]),ZERO1([illegible]),ZERO2([illegible]),[illegible]([illegible])
C..[illegible] IS THE [illegible] OF [illegible] [illegible],[illegible] IS THE [illegible] OF [illegible]
C..DIMENSIONING   POLE([illegible]),POLEI([illegible]) ZERO([illegible]),ZEROI([illegible])...WHERE,[illegible]
C..THE VECTOR X STORES [illegible] FROM 1 TO [illegible],[illegible] FROM K TO 2*K ..ETC
C..THE FILTER SPECIFICATIONS IS TO BE WRITTEN BETWEEN THE DOTTED LINE
C.................................................................................
      [illegible]=30
      K=1
      L=3
C
C  L=NUMBER OF [illegible] [illegible] [illegible] [illegible] [illegible] CONVERGE
C
      X(1)=0.0
      X(2)=0.0
      X(3)=0.0
      X(4)=-0.25
      YD(1)=1.0D0
      W(1)=0.D0
      DO 1 I=1,10
      YD(I)=1.0
    1 W(I+1)=W(I)+0.01D0
      YD(11)=0.5
      W(11)=0.1D0
      DO 2 I=12,21
      YD(I)=0.D0
    2 W(I)=W(I-1)+0.01
      DO 3 I=22,30
      YD(I)=0.0
    3 W(I)=W(I-1)+0.1
      LIMIT=100
C.................................................................................
      M=4*K
      DO 10 I=1,M
      CP=DCMPLX(0.D0,W(I)*PY)
   10 Z(I)=CDEXP(CP)
      EST=0.1
      EPS=1.0/10.**5
      ITER=1
  201 CONTINUE
      CALL DEMEP(FUNCT,M,X,O,DO,EST,EPS,LIMIT,IER,HC)
      LIMIT=25
      WRITE(6,11) IER
   11 FORMAT(5X,'IER=',I3)
      WRITE(6,200) KOUNT
```

```
      WRITE(6,14)Q
   14 FORMAT(5X,'Q=',D20.12)
      DO 23 I=1,K
      K1=K+I
      K2=2*K+I
      K3=3*K+I
      A(I)=X(I)
      B(I)=X(K1)
      C(I)=X(K2)
      D(I)=X(K3)
      R1=DCMPLX(C(I)**2-4.0*D(I),0.D0)
      R2=DCMPLX(A(I)**2-4.0*B(I),0.D0)
      POLE1(I)=(-C(I)+CDSQRT(R1))/2.0
      POLE2(I)=(-C(I)-CDSQRT(R1))/2.0
      P1(I)=CDABS(POLE1(I))
      P2(I)=CDABS(POLE2(I))
      ZERO1(I)=(-A(I)+CDSQRT(R2))/2.0
      ZERO2(I)=(-A(I)-CDSQRT(R2))/2.0
      Z1(I)=CDABS(ZERO1(I))
      Z2(I)=CDABS(ZERO2(I))
      IF(P1(I)-1.D0)15,15,16
   16 POLE1(I)=ANGLE(POLE1(I))/P1(I)
   15 IF(P2(I)-1.D0)17,17,18
   18 POLE2(I)=ANGLE(POLE2(I))/P2(I)
   17 IF(Z1(I)-1.D0)19,19,20
   20 ZERO1(I)=ANGLE(ZERO1(I))/Z1(I)
   19 IF(Z2(I)-1.D0)21,21,22
   22 ZERO2(I)=ANGLE(ZERO2(I))/Z2(I)
   21 A(I)=-(ZERO1(I)+ZERO2(I))
      B(I)=ZERO1(I)*ZERO2(I)
      C(I)=-(POLE1(I)+POLE2(I))
      D(I)=POLE1(I)*POLE2(I)
      X(I)=A(I)
      X(K1)=B(I)
      X(K2)=C(I)
      X(K3)=D(I)
   23 CONTINUE
      ITER=ITER+1
      IF(ITER.GT.L)GO TO 202
      DO 100 I=1,K
      IF(Z1(I).GT.1.D0) GO TO 201
      IF(Z2(I).GT.1.D0) GO TO 201
      IF(P1(I).GT.1.D0) GO TO 201
      IF(P2(I).GT.1.D0) GO TO 201
  100 CONTINUE
  202 CONTINUE
      CALL FUNCT(N,X,Q,DQ)
      WRITE(6,12) A0
   12 FORMAT(5X,'A0=',D20.12)
      DO 24 I=1,K
   24 WRITE(6,25)I,POLE1(I),I,POLE2(I),I,ZERO1(I),I,ZERO2(I)
   25 FORMAT(5X,'POLE1(',I3,')=',D20.12,4X,D20.12/5X,'POLE2(',I3,')='
     1 D20.12,4X,D20.12/5X,'ZERO1(',I3,')=',D20.12,4X,D20.12/5X,
     1 5X,'ZERO2(',I3,')=',D20.12,4X,D20.12)
      DO 33 I=1,K
   33 WRITE(6,32)I,A(I),I,B(I),I,C(I),I,D(I)
   32 FORMAT(5X,'A(',I3,')=',D20.12/5X,'B(',I3,')=',D20.12/5X,'C(',I3
     1 ')=',D20.12/5X,'D(',I3,')=',D20.12)
      DO 26 I=1,N
      Y(I)=A0*CDABS(H(I))
      ARG=AIMAG(H(I))/DREAL(H(I))
```

```
26    PHASE(I)=DATAN(ARG)*180./PY
      WRITE(6,27)
27    FORMAT(5X,'I',20X,'Y(I)',2?X,'W(I)',22X,'PHASE(I)')
      WRITE(6,28)(I,Y(I),W(I),PHASE(I),I=1,N)
28    FORMAT(4X,I3,9X,D20.12,7X,D20.12,10X,D20.12)
      DO 29 I=1,N
      YDS(I)=SNGL(YD(I))
      WS(I)=SNGL(W(I))
      YS(I)=SNGL(Y(I))
29    PHAS(I)=SNGL(PHASE(I))
      WRITE(6,30)
30    FORMAT(1H1,'PLOTTING OF THE GIVEN FILTER MAGNITUDE RESPONSE')
      CALL PLOT3(WS,YDS,N)
      WRITE(6,31)
31    FORMAT(1H1,'PLOTTING OF THE DESIGNED FILTER MAGNITUDE RESPONSE')
      CALL PLOT3(WS,YS,N)
      WRITE(6,101)
101   FORMAT(1H1,'PLOTTING OF THE FILTER PHASE RESPONSE')
      CALL PLOT3(WS,PHAS,N)
      DO 500 I=1,K
      KI=K+I
      POLER(I)=SNGL(DREAL(POLE1(I)))
      POLER(KI)=SNGL(DREAL(POLE2(I)))
      POLEI(I)=SNGL(AIMAG(POLE1(I)))
      POLEI(KI)=SNGL(AIMAG(POLE2(I)))
      ZEROR(I)=SNGL(DREAL(ZERO1(I)))
      ZEROR(KI)=SNGL(DREAL(ZERO2(I)))
      ZEROI(I)=SNGL(AIMAG(ZERO1(I)))
500   ZEROI(KI)=SNGL(AIMAG(ZERO2(I)))
      M=2*K
      CALL ZPLOT(POLER,POLEI,ZEROR,ZEROI,M)
      K=K+1
      LIMIT=100
      X(8)=X(4)
      X(7)=X(4)
      X(6)=X(3)
      X(5)=X(3)
      X(4)=X(2)
      X(3)=X(2)
      X(2)=X(1)
      IF(K.EQ.2) GO TO 201
      STOP
      END
```

[illegible]TRY

```
      SUBROUTINE FUNCT(P,X, [illegible])
      [illegible]
      DOUBLE PRECISION X([illegible]),[illegible],SUM1,SUM2,F(30),S1,S2,S3,S4,[illegible]
     1 YP(30),PREAL,CPANS
      COMPLEX*16 [illegible],A1(2,30),A2(2,30),[illegible](30), Z(30),DCMPLX,A1,A2,A3, [illegible]
      COMMON /P1/Z,[illegible]
      COMMON /P2/YP,[illegible]
      COMMON /P3/[illegible],
      SUM1=0.D0
      SUM2=0.D0
      [illegible]=0.[illegible]
      DO 2 J=1,
      [illegible]=DCMPLX(1.[illegible],0.DC)
      DO [illegible] I=1,[illegible]
      [illegible]1=[illegible]+I
      [illegible]=2*[illegible]+I
      [illegible]=3*[illegible]+I
      A1(I,J)=Z(J)**2+[illegible](I)*Z(J)+[illegible]([illegible])
    1 A2(I,J)=Z(J)**2+[illegible]([illegible])*Z(J)+X([illegible])
      [illegible]=[illegible]([illegible]A1(I,J)/A2(I,J))
      [illegible](J)=[illegible]
      SUM1=SUM1+[illegible]S([illegible](J))*[illegible](J)
    2 SUM2=SUM2+[illegible]S([illegible](J))**2
      [illegible]=SUM1/SUM2
      DO [illegible] J=[illegible],
      F(J)=[illegible]S([illegible](J))-[illegible](J)
    5 [illegible]=[illegible]+F(J)**[illegible]
      DO 7 J=[illegible],
      [illegible]=[illegible]+I
      [illegible]=2*[illegible]+I
      [illegible]=3*[illegible]+I
      S1=0.[illegible]
      S2=0.D0
      S3=0.D0
      S4=0.D0
      DO [illegible] J=1,
      [illegible]L=2.[illegible]*F(J)*[illegible]S([illegible](J))
      [illegible]=Z(J)/A1(I,J)
      [illegible]2=1./A1(I,J)
      [illegible]=Z(J)/A2(I,J)
      [illegible]=1./A2(I,J)
      S1=S1+[illegible]L*[illegible]L([illegible]1)
      S2=S2+[illegible]L*[illegible]L([illegible]2)
      [illegible]=[illegible]-[illegible]L*[illegible]L([illegible])
    6 S4=S4-[illegible]L*[illegible]L([illegible])
      [illegible](I)=S1
      [illegible]([illegible]1)=S2
      [illegible]([illegible]2)=S3
    7 [illegible]([illegible])=S4
      [illegible]
      [illegible]
```

```
      $ENTRY



      COMPLEX FUNCTION ANGLE*16(X)
      DOUBLE PRECISION THETA,DATAN,DIMAG,DREAL,DSIN,DCOS
      COMPLEX*16 X,DCMPLX
      THETA=DATAN(DIMAG(X)/DREAL(X))
      ANGLE=DCMPLX(DCOS(THETA),DSIN(THETA))
      RETURN
      END



      DOUBLE PRECISION FUNCTION AIM(X)
      COMPLEX*16 X,P,DCONJG,DCMPLX
      P=DCMPLX(0.D0,1.D0)
      AIM=(X-DCONJG(X))/(2.D0*P)
      RETURN
      END
```

```
        SENTRY

C DESIGN OF DIGITAL FILTERS BY IMPLEMENTING CONSTRAINTS ON AN ANALOG FI
      DIMENSION A(2),B(2),C(2),D(2),E(2),F(2),G(2),COST(30),Y(30),
     1 Y(30),HC(60) ,POLER(4),POLEI(4),ZEROR(4),ZEROI(4)
      COMPLEX Z(30),CP,P1,P2,POLE1(2),POLE2(2),ZERO1(2),ZERO2(2),H(3
     1 CMPLX,CEXP,CSQRT
      COMMON/B1/Z,H
      COMMON/B3/AP,YD
      COMMON/B4/W,K
      COMMON/B2/KOUNT
      EXTERNAL FUNCT
C..FILTER SPECIFICATIONS
C
      PY=3.141592
      K=1
      N=27
      Y(1)=1.414
      Y(2)=1.0
      Y(3)=1.02
      Y(4)=1.0
      YD(1)=1.0
      W(1)=0.
      DO 3 I=1,10
      YD(I)=1.
    3 W(I+1)=W(I)+0.01
      YD(11)=0.5
      W(11)=0.1
      DO 12 I=12,21
      YD(I)=0.
   12 W(I)=W(I-1)+0.01
      DO 13 I=22,27
      YD(I)=0.
   13 W(I)=W(I-1)+0.1
C
C..EVALUATION OF Z(I)
      DO 100 I=1,N
      CP=CMPLX(0.0,W(I)*PY)
  100 Z(I)=CEXP(CP)
C
      EST=0.1
      EPS=1.0/10.0E8
      LIMIT=100
      M=4*K
      CALL FLEPR(FUNCT,M,X,F,GG,EST,EPS,LIMIT,IER,HC)
      WRITE(6,200) KOUNT
  200 FORMAT(5X,'KOUNT=',I5)
      WRITE(6,101)IER,F
  101 FORMAT(5X,'IER=',I3/5X,'F=',E12.5)
      WRITE(6,400) AP
  400 FORMAT(5X,'ANALOG FILTER GAIN=',F12.5)
      WRITE(6,102)(I,Y(I),I=1,M)
  102 FORMAT(5X,'X(',I3,')=',F12.5)
      WRITE(6,201)
  201 FORMAT(5X,'IF THE DIGITAL FILTER HAD THE FORM'/ 7X,'Y(Z)=A0*(
     1 UCT  FROM I=1 TO K OF, (Z**2+A(I)*Z+B(I))/(Z**2+C(I)*Z+D(I)),THE
     1 )
      DO 107 I=1,N
      Y(I)=CABS(AP*H(I))
  107 PHASE(I)=ATAN(AIMAG(H(I))/REAL(H(I)))
      A1=1.0
```

```
      K1=K+I
      K2=2*K+I
      K3=3*K+I
      A(I)=(-2.0+2.0*Y(K1)**2)/(1.0+X(I)**2+Y(K1)**2)
      B(I)=(1.0-X(I)**2+X(K1)**2)/(1.0+X(I)**2+Y(K1)**2)
      C(I)=(-2.0+2.0*Y(K2)**2)/(1.0+Y(I)**2+Y(K1)**2)
      [illegible](I)=(1.0-[illegible](K2)[illegible])/(1.0+X(K2)**2+Y(K3)**2)
      F1=CMPLX(C(I)**2-4.0*D(I),0.)
      F2=CMPLX(A(I)**2-4.0*B(I),0.)
      POLE1(I)=-C(I)/2.0+CSQRT(F1)/2.0
      POLE2(I)=-C(I)/2.0-CSQRT(F1)/2.0
      ZERO1(I)=-A(I)/2.0+CSQRT(F2)/2.0
      ZERO2(I)=-A(I)/2.0-CSQRT(F2)/2.0
      A1=[illegible]*(1.0[illegible]*Y(I)**2+Y(K1)**2)/(1.0+X(K2)**2+Y(K3)**2)
      WRITE(6,105)I,A(I),I,B(I),I,C(I),I,D(I)
  105 FORMAT(5X,'A(',I2,')=',F12.5/5X,'B(',I2,')=',F12.5/5X,'C(',I2,
     1')=',F12.5/5X,'D(',I2,')=',F12.5)
      WRITE(6,106)I,POLE1(I),I,POLE2(I),I,ZERO1(I),I,ZERO2(I)
  106 FORMAT(5X,'POLE1(',I3,')=',F12.5,5X,F12.5/5X,'POLE2(',I3,')=',
     1 F12.5,2X,F12.5/5X,'ZERO1(',I3,')=',F12.5,5X,F12.5/5X,'ZERO2(',I3,
     1 ')=',F12.5,2X,F12.5)
  104 CONTINUE
      [illegible]
      WRITE(6,107)[illegible]
  107 FORMAT(5X,'[illegible]=',F12.5)
      WRITE(6,10[illegible])
  10[illegible] FORMAT(1H1,5X,'PLOTTING OF THE GIVEN FILTER [illegible]')
      CALL PLOTP([illegible],Y[illegible])
      WRITE(6,108)
  108 FORMAT(1H1,5X,'PLOTTING OF THE DESIGN OF FILTER [illegible]
     1')
      CALL PLOTP([illegible],Y,[illegible])
      [illegible](Y(I),[illegible](I),[illegible](I),[illegible](I),I=1,[illegible])
  110 FORMAT(4F12.5)
      DO [illegible] I=1,[illegible]
      [illegible]=[illegible]+I
      POLE[illegible](I)=REAL(POLE1(I))
      POLE[illegible](KI)=REAL(POLE2(I))
      POLEI(I)=AIMAG(POLE1(I))
      POLEI(KI)=AIMAG(POLE2(I))
      ZERO[illegible](I)=REAL(ZERO1(I))
      ZERO[illegible](KI)=REAL(ZERO2(I))
      ZERO[illegible](I)=AIMAG(ZERO1(I))
  500 ZERO[illegible](KI)=AIMAG(ZERO2(I))
      [illegible]
      CALL SPLOT([illegible])
      [illegible]=[illegible]+1
      X([illegible])=X(4)
      Y([illegible])=X(4)
      Y(4)=X(3)
      Y(3)=X(3)
      X(4)=X(2)
      X(3)=X(2)
      Y(2)=X(1)
      IF([illegible].2)GO TO 1
      STOP
      END
```

```
      SUBROUTINE FUNCT(N,X,Q,DQ)
      DIMENSION X(N),DQ(N),YD(30),F(30)
      COMPLEX Z(30),RN(2,30),RD(2,30),H(30),P,Z1,Z2,Z3,Z4,CMPLX
      COMMON/B1/Z,H
      COMMON/B3/AQ,YD
      COMMON/B4/M,K
      Q=0.0
      SUM1=0.0
      SUM2=0.0
      DO 2 J=1,M
      P=CMPLX(1.0,0.0)
      DO 1 I=1,K
      K1=K+I
      K2=2*K+I
      K3=3*K+I
      RN(I,J)=(Z(J)**2)*(1.0 +(X(I)**2)+(X(K1)**2))+Z(J)*(2.0*(X(K1)**
     1 )-2.0)+(1.0-(X(I)**2)+(X(K1)**2))
      RD(I,J)=(Z(J)**2)*(1.0+(X(K2)**2)+(X(K3)**2))+Z(J)*((2.0*
     1 X(K3)**2)-2.0)+(1.0-(X(K2)**2)+(X(K3)**2))
    1 P=P*(RN(I,J)/RD(I,J))
      H(J)=P
      SUM1=SUM1+CABS(H(J))*YD(J)
    2 SUM2=SUM2+CABS(H(J))**2
      AQ=SUM1/SUM2
      DO 3 J=1,M
      F(J)=AQ*CABS(H(J))-YD(J)
    3 Q=Q+F(J)**2
      DO 4 I=1,K
      K1=K+I
      K2=2*K+I
      K3=3*K+I
      S1=0.
      S2=0.
      S3=0.
      S4=0.
      DO 5 J=1,M
      PL=2.*AQ*F(J)*CABS(H(J))
      Z1=(2.*X(I)*(Z(J)**2)-2.*X(I))/RN(I,J)
      Z2=(2.*X(K1)*(Z(J)**2)+4.*X(K1)*Z(J)+2.*X(K1))/RN(I,J)
      Z3=-(2.*X(K2)*(Z(J)**2)-2.*X(K2))/RD(I,J)
      Z4=-(2.*X(K3)*(Z(J)**2)+4.*X(K3)*Z(J)+2.*X(K3))/RD(I,J)
      S1=S1+PL*REAL(Z1)
      S2=S2+PL*REAL(Z2)
      S3=S3+PL*REAL(Z3)
    5 S4=S4+PL*REAL(Z4)
      DQ(I)=S1
      DQ(K1)=S2
      DQ(K2)=S3
    4 DQ(K3)=S4
      RETURN
      END
```

VITA AUCTORIS

1945 Born on the 24th of March in Alexandria, Egypt.

1963 Matriculated from Victoria College , Alexandria, Egypt.

1968 Graduated from the University of Alexandria with a B.Sc. degree in Electrical Engineering.

1970 Registered as a Candidate for M.A.Sc. in Electrical Engineering at the University of Windsor.